

OTDK DOLGOZAT

Apró Fanni

2024.



SZEGEDI TUDOMÁNYEGYETEM
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
MÉRNÖKI KAR

***HŰTŐGÉPHÁZ PLC ÉS SCADA RENDSZERÉNEK
KORSZERŰSÍTÉSE***

OTDK DOLGOZAT

Apró Fanni

mechatronikai mérnök BSc

hallgató

Témavezető:

Csikós Sándor

tanársegéd

SZEGED

2024.

TARTALOMJEGYZÉK

1. BEVEZETÉS	5
2. IRODALMI ÁTTEKINTÉS.....	6
2.1. Hűtéstechnika	6
2.1.1. <i>Mesterséges hűtés fejlődésének áttekintése</i>	<i>6</i>
2.1.2. <i>Hűtés szerepe az élelmiszeriparban</i>	<i>7</i>
2.1.3. <i>Hűtőkörfolyamatok</i>	<i>9</i>
2.1.4. <i>Hűtőközegek</i>	<i>11</i>
2.1.5. <i>Hűtéshez használt berendezések</i>	<i>12</i>
2.1.6. <i>A hűtőberendezések automatizálásának kialakulása.....</i>	<i>16</i>
2.2. Ipari automatizálás	17
2.2.1. <i>Az automatizálási piramis</i>	<i>18</i>
2.2.2. <i>PLC rendszerek.....</i>	<i>20</i>
2.2.3. <i>SCADA rendszerek.....</i>	<i>23</i>
3. ANYAGOK ÉS MÓDSZEREK	25
3.1. Hardveres eszközök.....	25
3.1.1. <i>Omron PLC.....</i>	<i>25</i>
3.1.2. <i>Érzékelők</i>	<i>26</i>
3.2. SCADA szoftver.....	26
3.3. Egyéb információ források	27
3.3.1. <i>Meglévő SCADA felület</i>	<i>27</i>
3.3.2. <i>Wizcon tag lista.....</i>	<i>27</i>
3.3.3. <i>Omron PLC program</i>	<i>28</i>
3.3.4. <i>Vezérlő szekrény kapcsolási rajza</i>	<i>30</i>
4. ÖNÁLLÓ MUNKAVÉGZÉS.....	31
4.1. <i>Célkitűzés megismétlése, elvárások ismertetése</i>	<i>31</i>
4.2. <i>PLC-n futó programrészek folyamatábrája</i>	<i>32</i>

4.2.1.	<i>Alap program.....</i>	32
4.2.2.	<i>Ammónia szivattyúk programja</i>	33
4.2.3.	<i>Hibajelek program.....</i>	34
4.2.4.	<i>Kondenzátor programja</i>	35
4.2.5.	<i>Olvasztóprogram.....</i>	35
4.2.6.	<i>Raktár, folyosók és üzemek programja.....</i>	36
4.3.	<i>Tagek kezelése</i>	37
4.3.1.	<i>Tagek felvétele a Gateway-en.....</i>	37
4.3.2.	<i>Tagek formázása a Designer-ben</i>	38
4.4.	<i>Megjelenítő felület elkészítése</i>	40
4.4.1.	<i>Komponens Binding-olása</i>	42
4.4.2.	<i>Komponens Skript-elése, Event-ek készítése</i>	44
4.5.	<i>Funkciók elkészítése</i>	45
4.5.1.	<i>Grafikonok kirajzolása.....</i>	45
4.5.2.	<i>Exportálások megalkotása</i>	47
4.5.3.	<i>Felhasználói szintek kezelése</i>	50
4.5.4.	<i>Alarm rendszer kialakítása</i>	51
5.	EREDMÉNYEK.....	53
6.	ÖSSZEFOGLALÁS	55
	IRODALOMJEGYZÉK.....	56

1. BEVEZETÉS

Egy magyarországi élelmiszeripari cég telephelyén lévő hűtőgépház vezérlése egy Omron PLC-n, és egy hozzá tartozó SCADA rendszeren futott. A SCADA régi és elavultnak számított, valamint csak és kizárólag Windows XP-s operációs rendszerű számítógépre telepíthető. A cég a Windows XP operációs rendszert már nem támogatta, így a hálózatra sem lehetett felcsatlakoztatni a számítógépet, hogy távolról is elérje a kezelő. A SCADA rendszer felhasználója csak akkor tudott módosítást végezni, és értesülni a hűtőgépház állapotáról, ha fizikálisan is a számítógép közelében volt. Esetleges meghibásodás esetén a SCADA nem telepíthető újra, hiszen a szükséges operációs rendszer már nem támogatott a cégcsoporton belül.

Ezen indokok miatt, szükségszerűnek láttam a SCADA rendszer korszerűsítését. A fejlesztés során a rendszer üzemeltető felhasználók további új igényei is beépítésre kerültek a rendszerbe. Ez lehetővé tette, hogy a felhasználók számára a rendszer a lehető leghatékonyabb felhasználást nyújtsa.

Munkám célja a felsorolt elvárások teljesítése a felhasználók és a vezetőség számára, valamint a cégben használt SCADA készítési belső standardok betartása (használt megnevezések, sémák stb.), hogy a későbbiekben könnyedén módosítható és átlátható legyen más cégben dolgozó kollégák számára.

Ezen célok eléréséhez szükségem volt a hűtéstechikai és ipari automatizálási alapok elsajátítására, a PLC-n futó program és az egész rendszer megértéséhez, valamint a SCADA tervező szoftver használatának megtanulására. Azért, hogy egy olyan működő és a felhasználók által kedvelt SCADA rendszert alkossak, mely a cégcsoport fő hálózatára csatlakozott eszközről elérhető legyen, telephelytől és eszköztől függetlenül. A régi rendszer kinézetének és funkcióinak megtartása mellett, kibővítettem és módosítottam az elvárások szerint, például: adatok tárolása és kiválasztott időintervallum szerint exportálása Excel és Pdf formátumban, mért értékek kirajzolása grafikonokra, felhasználói szintek kezelése, riasztások esetén e-mail értesítés a felhasználók számára.

2. IRODALMI ÁTTEKINTÉS

A fejezetben ismertetem a dolgozat elkészítéséhez szükséges elméleti alapokat, melyek között szerepel a hűtéstechnikai alapismeretek, és a hozzá tartozó berendezések. Valamint az ipari automatizálás alapjait, hangsúlyt fektetve a PLC és SCADA rendszerek működésére.

2.1. Hűtéstechnika

A köznyelvben használt „hideg”, a fizikában helyes megfogalmazás szerint „alacsonyabb hőmérsékletű meleg”, azaz egy anyag molekuláinak kisebb mozgási energiája. Fizikai szemmel nézve hűtés akkor jön létre, amikor egy nála kisebb hőmérsékletű közeggel hőt vonunk el, vagyis a hidegebb állapot lényegében egy elvont hőmennyiséggel csökkentett meleg állapot. Hűtőgép pedig olyan műszaki berendezés, mely egy anyagból hőt von el (Jakab, 2006).

A hő természetes módon a magasabb hőmérsékletű helytől az alacsonyabb hőmérsékletű helyre áramlik. Hűtés során ezt az áramlási irányt fordítjuk meg, és ez a művelet energia befektetéssel jár. Ez az energia lehet munka (kompresszoros hűtőgépek), hő (abszorpciós hűtőgépek), de akár közvetlenül elektromos energiával is meg lehet valósítani a hűtést. A hűtőgépeknek általában szükségük van hűtőközegre, mely az alacsonyabb helyről felveszi a szállítandó hőmennyiséget, majd a magasabb hőmérsékletű helyen adja le. A kompresszoros hűtés használata és ammónia, mint hűtőközeg alkalmazása a legelterjedtebb (Rácz, 2011a).

2.1.1. *Mesterséges hűtés fejlődésének áttekintése*

A hűtés iránti igényről, annak kielégítési módjáról már a régi időkből maradtak fent írásos emlékek. Hosszú időn keresztül a természetes jeget és havat használták az élelmiszerek, italok vagy akár lakott helyiségek hűtésére. Ez a módszer viszont nagyon korlátozott volt a természetes hűtőanyagok előfordulása miatt, a jéggel való kereskedelem ugyan felvirágzott, de a hűtés igény növekedése megkövetelte a technológia fejlesztését is.

Egy olyan hűtési módra volt szükség, mely földrajzi fekvéstől, évszaktól, időjárástól független. A mesterséges hűtés kialakulásának feltétele a kellő technológiai fejlettség volt, melyre a 19. század lehetőséget nyújtott. 1834-ben Jakob Perkins szabadalmaztatta az első etil-éter hűtőközeggel működő kompresszoros hűtőberendezést, viszont a berendezés robbanásveszélyesnek bizonyult. Az első iparilag is hasznosított jéggyártó hűtőberendezést 1859-ban Carre fejlesztette ki. Megalkotta az abszorpciós hűtőberendezést, majd 1867-ben elsőként alkalmazta az ammóniát kompresszoros berendezés hűtőközegeként (Láng & Jakab, 1984).

A mesterséges hűtés problémájának megoldása, lehetővé tette, az alkalmazási területek bővülését. Kezdetben élelmiszeripari termékek hűtésére szolgált, valamint jéggyártásra. Majd a termékek hűtött szállítása is elkezdődött. Ezután elkezdődött a hűtő- és fagyasztóházak létesítése. Az alkalmazási lehetőségek bővülésével arányosan fejlődött a hűtés technológiája is. A korabeli hűtőberendezések azonban a maiakkal összehasonlítva hatalmas helyet foglaltak, és folyamatos felügyeletet igényeltek, ez miatt gazdaságosan csak raktárak és jéggyárakban lehetett használni. Ma már a mesterséges hűtés alkalmazása az élet minden területén körülvesz minket, kezdve különféle produktumok előállításán át, a háztartástól a hatalmas ipari gyárkomplexumok részéig (Gänger, 1964).

2.1.2. *Hűtés szerepe az élelmiszeriparban*

A hűtéstechnikának az élelmiszeriparban két alapvető feladatot kell megoldania. Az egyik a legkedvezőbb hőmérsékleti feltételek megteremtése a különféle élelmiszerek előállításához és feldolgozásához. A másik fontos feladat pedig az értékes termékek tárolóképességének meghosszabbítása. A hűtési mód megválasztása a hűtendő árutól, az alapanyagtól vagy a termék előállítási folyamattól függ (Beke, 1994).

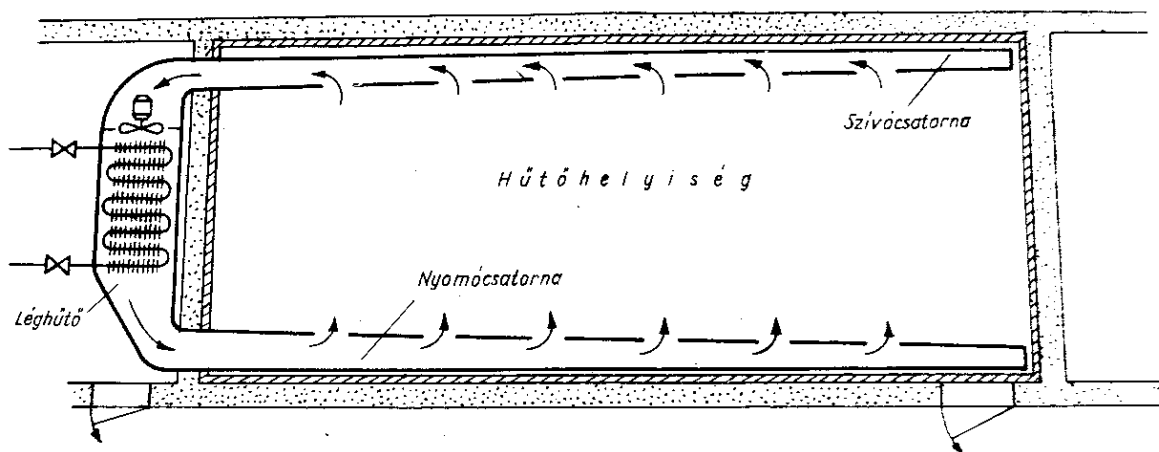
Tuhsnajt és Emblik (1956) könyvében olvashatunk a különféle élelmiszerek hűtési és fagyasztási eljárásairól. Írományaik rávilágítanak arra, hogy a hűtés módjának megválasztása, értsd ezalatt a megfelelő hőmérsékletet, és e hőmérséklet elérésének idejét, a hűtendő alapanyagtól vagy terméktől függ. A friss vágásból származó hús a mikroorganizmusok elszaporodásának igen jó lehetőséget nyújt. A hőmérséklet csökkentése és a felület leszárlásával ez a szaporodási folyamat meggátolható. Viszont a hús fagyasztása előtt egy előhűtési folyamatra van szükség. Ez két szempontból is fontos: a húsban történő érés¹, szükséges része meg tud történni, valamint a fagyasztóhelyiség hőmérsékletét károsan növelné az előhűtés nélkül betárolt hús. A vaddal és a baromfival hasonló a helyzet, ellentétben a halfélékkel, melyeknek a testhőmérséklete alig tér el a környezetétől. A halfélék esetében a 0 °C-hoz közeli gyors lehűtésre kell törekedni. A tojás, esetében, már a páratartalom is fontos szempont a hűtés során. Ezt a terméket lassan kell lehűteni, és igyekezni a páratartalom alacsony szinten tartásához. Ezen példákból is jól látható, hogy a hűtendő termék határozza meg a hűtési folyamat kialakítását, tehát a hűtőházak és élelmiszert feldolgozó üzemek hűtőrendszerének megtervezése előtt, a legfontosabb szempont felmérni az adott terméket, hogy milyen igényekkel rendelkezik.

¹ Az érés a húsban történő vegyi folyamatok sorozata, mely végbemenetele után lesz teljesen élvezhető., viszont túl nagy hőmérsékleten túl gyors az érési folyamat és a hús megromlik.

Hűtési módok az élelmiszeriparban

Az élelmiszeriparban használt hűtési módok közül a legelterjedtebb a levegőben történő hűtés. Ekkor az árut a hűtőhelyiségben az erre alkalmas berendezésekkel elhelyezzük (állványzat, húsoshorgok stb.), és a megfelelő hűtőkészülékkel hűtjük. Az élelmiszer a nála kisebb hőmérsékletű levegővel kerül érintkezésbe, és ennek során lehül, a tovább áramló levegő pedig felmelegszik. A módszer előnye, hogy a termék természetes közegben marad, idegen anyagok behatása nélkül (Beke, 1994).

A levegővel való hűtés történhet csendes, vagy légmozgatásos módon. Légmozgatásos hűtés esetén, egy kültéri léghűtő berendezés helyezkedik el a termékkel felszerelt kamrán kívül, és a levegő mozgásáért ventilátorok felelősek. A lehűtött levegő a beáramló csöveken keresztül a térben érintkezik a termékekkel és a szívóoldalon felhelyezett csöveken keresztül távozik a térből, majd visszaérkezik a léghűtőberendezésbe (1. ábra). Ezzel az eljárással a helyiség egyenletesen áthűthető. Sok esetben használatos még a beltéri léghűtő készülék, mely szintén a hűtőtestből és ventilátorokból áll. Viszont ez egy kisebb, közös burkolattal rendelkező összeépített egység, melyet a hűteni kívánt helyiség belső falára szerelnek fel, adott esetben hűtésigény és a tér nagyságától függően akár többet is. Nagy népszerű kialakítás, az úgynevezett hűtőalagutak, melyek hosszúka, téglatest alakú, alacsony belmagasságú kamrák. Ezekben a kamrákban nagy sebességgel hideg levegőt fúvatnak keresztül a betárazott termékek között (Gänger, 1964).



1. ábra: Léghűtőkészülék (Gänger, 1964)

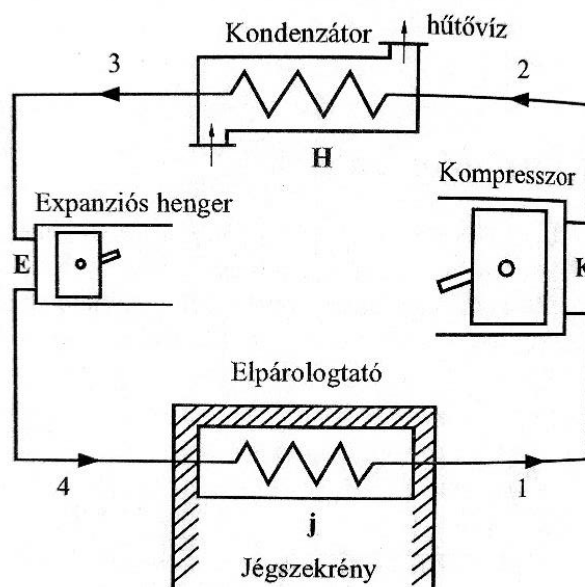
A levegővel való hűtést alkalmazzák általában hús, baromfi, vad, tojás, gyümölcs és zöldség hűtésére. De ezenkívül halfélék hűtésére gyakran használnak vizet, aprított jeget és sólét. Az aprított jéggel való hűtés mai napig a baromfiiparban is jelen van. A tej lehűtésére különleges készülék szolgál, melyben speciális hűtőfolyadék kering. Gyümölcsök hűtési eljárásában gyakran előfordul, hogy az atmoszférát különféle gázokkal dúsítják (CO_2 és O_2), illetve a

gyümölcsök és a zöldségek hűtési eljárásában megjelent a pelyhesített jég szórása (Tuhsnajt & Emblik, 1956).

2.1.3. Hűtőkörfolyamatok

„A hűtőkörfolyamat a hűtőközeg állapotváltozásainak azon egymásutáni sorozata, amelyek során a hűtőközeg felveszi a hűtendő közegből elszállítandó hőt, majd külső munka (energia) befektetése révén hőmérséklete az atmoszférikus szintre emelkedik, ahol hőt ad le a hőt felvevő közegnek és hasznos munkavégzés mellett, vagy anélkül, hőmérséklete a hőfelvétel (hűtés) hőmérséklet szintjére kerül.” (Beke, 1994)

A hűtőberendezések alapvetően azt a fizikai törvényt használják ki, hogy a párolgó levegő hűti a környezetét. A hűtő-körfolyamat, vagyis elpárolgás és kondenzáció folyamatát a 2. ábra szemlélteti.



2. ábra: A hűtési körfolyamat elvi sémája (Beke, 1994)

Az elpárolgatóban keletkezett gőzt nem hagyjuk elveszni, hanem egy kondenzátornak nevezett, nagy felületű hőcserélőn keresztül újra visszakényszerítjük folyadék halmazállapotba (2. ábra (2)). Ez a folyamat úgy jön létre, hogy a hőcserélő nagy felületének köszönhetően hűti az átfolyó közeget, ennek hatására a gőz lehűl, és lekondenzálódik, vagyis ismét folyékony halmazállapotba kerül. A kondenzátort elhagyva egy adagolóba kerül, melynek nagyon kicsi a keresztmetszete, és ezen áthaladva a közeg nyomása csökken (2. ábra (3)). A nyomás csökkenésével csökken a hőmérséklet is, mivel a két állapotjelző a telítettségi állapot² szigorúan

² A telített állapotban a hűtőközeg gőz formájában van, és már nem tud több hőt felvenni anélkül, hogy kondenzálna.

összetartozó értékpárja. Az adagoló másik feladata, hogy az elpárologtatóba megfelelő mennyiségű közeget enged beáramolni. Az elpárologtatóban, újra elpárolog az alacsony nyomású és hőmérsékletű közeg a meleg levegő hatására, és gőz halmazállapotúvá válik (2. ábra (4)). Ezt a gőzt szívja be a kompresszor, ami a hűtőközeg szállításán kívül annak nyomásemelkedését is okozza, és továbbítja újra a kondenzátor felé (2. ábra (1)). A halmazállapot- és nyomásváltozások ilyen körfolyamaton keresztül mennek végbe és a kompresszor működtetéséig folyamatosan ismétlődnek (Rác, 2011a).

A hűtőberendezésnek a működése során tehát a következő négy megfelelő sorrendű feladatot kell ellátni:

1. Adiabatus kompresszió: a gőz állapotú munkaközeg nyomásának és hőmérsékletének megnövelése. A rendszer és a környezete között hőátadás nem történik.
2. Izotermikus hőleadás: a munkaközeg kondenzációja, vagyis magas hőmérsékleten történő cseppfolyósítása.
3. Adiabatus expanzió: a folyadék halmazállapotú hűtőközeg nyomásának, és ezáltal hőmérsékletének csökkentése hőátadás nélkül.
4. Izotermikus hőfelvétel: az állandó, alacsony hőmérsékleten lejátszódó folyamat során a munkaközeg elpárolog és hőt von el a környezetétől, vagyis a hűtött közegtől (Vasáros, 2020).

Beke (1994) könyvében olvashatunk, a kompresszoros hűtőberendezésének kialakítási módjairól, amiből jól látható, hogy az előbbieken tárgyalt hűtőkörfolyamat eredménye többféle megoldással kialakítható és akár a hatásfoka növelhető. A fojtószelep alkalmazása a körfolyamatban, 2. ábrán látható 3 folyamatnál, vagyis az adiabatus expanziónál, igen hatásos, mivel a gyakorlatban használt nagy helyigényű és költséges expanziós henger, könnyedén helyettesíthető egy a csőhálózatba épített fojtószeleppel. A fajlagos hűtőteljesítmény növelésének egyik lehetősége, az utóhűtő alkalmazása, amely a kondenzátor után kerül a körfolyamatban beépítésre. Többfokozatú kompresszió alkalmazása nagy nyomásviszonynál szokás alkalmazni, a kompressziómunka csökkentése érdekében. Ezen példákon kívül még sok más kialakítási módja van a kompresszorral való hűtőkörfolyamatoknak, ám Beke (1994) felhívja a figyelmet arra a fontos tervezési szempontra is, hogy a valóságos körfolyamat több vonatkozásban eltér az előzőekben tárgyalt elméleti folyamatoktól, hiszen természetes jelenség, hogy minden folyamat veszteséggel jár.

Zárt hűtési körfolyamatot, nem csak mechanikai munka befektetésével, vagyis kompresszor használatával lehet fenttartani, hanem hőenergia felhasználásával is meglehet valósítani, vagyis abszorpciós hűtőberendezéssel. Ezen berendezések energiaköltséget tekintve

gazdaságtalanabbak, és hűtővízfogyasztásuk is 2-2,5-szerese a kompresszoros hűtőberendezésnek, tehát a kompresszoros berendezések használata az esetek nagy részében jobban megéri. Olyan helyen éri meg alkalmazni az abszorpciós berendezésnek, ahol hulladékhő hasznosítása a cél, vagy korlátlan mennyiségű olcsó hűtővíz áll rendelkezésre, mint például folyók vagy tavak mentén (Rácz, 2011a).

2.1.4. Hűtőközegek

Hűtőközeget egy olyan közeget értünk, amely alacsony hőmérsékleten elpárologva hőt von el és azt magas hőmérsékleten cseppfolyósodva leadja. Kompressziós körfolyamatokban a hűtőközeg zárt rendszerben kering, és a folyton ismétlődő körfolyamat során kétszer változtatja meg halmazállapotát (elpárolog és cseppfolyósodik), a hőmérsékletét (felmelegszik és lehül) és a nyomását (kiterjeszkedik és összenyomódik).

Ahhoz, hogy egy közeget hűtőközegként használhassunk, alacsony forrásponttal kell rendelkeznie, valamint nagy párolgáshővel és kis hőmérsékleten kicsiny fajlagos térfogattal. Ezek a tulajdonság teszik lehetővé, a berendezés kedvező méretben legyen kivitelezhető. Ezen fizikai jellemzőkhöz hozzátartozik még egy fontos adat, hogy azonos hűteljesítményhez szükséges kompresszorhenger térfogata hogyan alakul a különböző hűtőközegek esetén. A legkisebb kompresszorhenger térfogat a széndioxid és az ammónia esetében van (Vasáros, 2020).

A hűtőközegek megnevezése és jelölésének kialakulása a 20. század második felében történt meg, ahol a vegyjel helyett az ISO ajánlására márkanévtől és vegyjeltől független áttértek a különleges betű- és számjelzés-kombinációra. A jelölés módjában R³ betű és az azt követő számozás, melynek logikája a hűtőközegekben leggyakrabban előforduló kémiai összetételre vezethető vissza (Láng & Jakab, 1984).

Ammónia (R717)

Láng és Jakab (1984) kiemelték, hogy az ammónia (vegyjel: NH₃, ISO jelölés, mint hűtőközeg: R717) a leggyakrabban használt hűtőközeg a következő tulajdonságai miatt: mérsékelt nyomásszintek, nagyszintű hűtőteljesítmény, emiatt kis térfogatáramok, kedvező gép- és vezetékmeretek érhetőek el, kedvező hőátadási tulajdonságok, szivárgása egyértelműen kimutatható, olajat nem oldja és olcsó.

³ R betű használata az angol refrigerant szóból ered, mely jelentése: hűtőközeg.

Az ammónia tiszta, vízmentes, cseppfolyós állapotban vízhez hasonlóan átlátszó folyadék. Kifejezetten mérgező és roncsoló hatású. Gőze könnyebb a levegőnél, szúrós szaga a berendezés legkisebb tömítési hibáját is elárulja. 15-27 % ammónia tartalmú levegő a nyílt lángtól meggyullad. Erre viszont igen kis esély, hiszen ilyen koncentráció akkor sem állítható elő, ha egy ipari nagyságrendű berendezés egész hűtőközegetöltése kiáramlik. 2 % ammóniatartalmú levegő belélegzése esetén pedig az emberi ott tartózkodást ellehetetleníti. A legtöbb közegre igaz, hogy belélegzésük roncsoló vagy mérgező hatású, épp ezért azon hűtőközegeket, melyek önmagukban szagtalanok, szagosító anyaggal látják el, hogy szivárgás esetén érzékelhető legyen, mielőtt még veszélyessé válna (Gänger, 1964).

2.1.5. Hűtéshez használt berendezések

A hűtéshez használt berendezések közé tartozik a kézi elzáró szelepektől kezdve, a különféle mérőműszereken át a kompresszorig minden elem, mely részt vesz a hűtőberendezés működésében. Ebben a fejezetben főként a legfontosabb egységeket ismertetem (melyek a 2. ábrán szereplő körfolyamatban szerepeltek), kiemelve azt, hogy milyen szerepet töltenek be a körfolyamatban és milyen elven működnek.

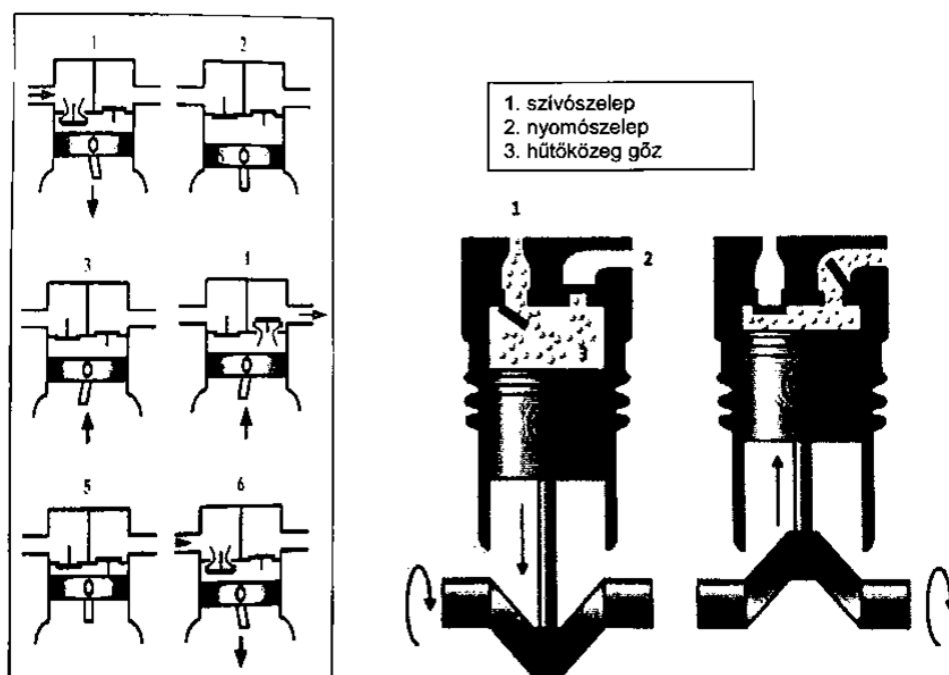
Kompresszorok

A kompresszor a körfolyamat fenntartását biztosító nyomást fokozó berendezés. A kompresszor szállítja a hőfelvétel követően a hűtőközegáramot a hőleadás magasabb nyomás- és hőmérséklet szintjére. A működéséhez szükséges energia fajtája szerint különbséget kell tenni kompresszoros berendezésekbe alkalmazott, mechanikai munkát igénylő úgynevezett mechanikus kompresszorok és a szorpciós berendezések termokémiai kompresszorai között. Az utóbbi esetén alapvetően csak megfelelő hőmérsékletű hőáramra van szükség.

Mechanikus kompresszorok esetén két típust különböztetünk meg, melyek a térfogat-kiszorításos és az áramlástechnikai elvén alapuló kompresszor kialakítások. Térfogat kiszorítás elvén alapuló kompresszorok közül elterjedt az alternáló dugattyús és a forgódugattyús kialakítás. Áramlástechnikai elven alapuló kompresszorok csoportját pedig a radiális és axiális turbókompresszorok alkotják (Láng & Jakab, 1984).

Alternáló dugattyús kompresszor munkafolyamatának bemutatását a 3. ábrán keresztül fogjuk megismerni. Az elő lépésben a dugattyú zárt szelepeknél szívóhatást hoz létre a hengertérben (3. ábra (1)), ennek hatására kinyílik a szívószelep és a hengertérbe a hűtőközeg-gőz áramlik be. Alsó holtponthoz elérése esetén (3. ábra (2)) a dugattyú mozgása megáll, és irányt vált, ekkor a szívószelep zárul. A 3. folyamat során történik a kompresszió (3. ábra (3)) a

dugattyú a zárt térben felfelé mozdul és összenyomja a közeget. Amikor a hengerben a nyomás meghaladja a nyomótérben uralkodó nyomást, vagyis a kondenzátornyomást, a nyomószelep kinyílik és a dugattyú kitolja az összesűrített közeget (3. ábra (4)). A Felső holtpont elérése esetén (3. ábra (5)) a dugattyú megáll, irányt vált és a nyomószelep bezárul. Az utolsó lépésben pedig a dugattyú lefelé történő mozgása közben (3. ábra (6)) a hengertérben a nyomás újra kiterjed és addig csökken, míg a szívócsatorna nyomását el nem éri (Jakab, 2006).



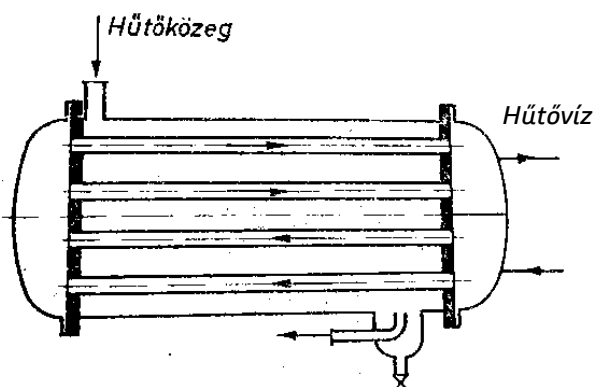
3. ábra: Dugattyús kompresszor munkafolyamata és elemei (Rácz, 2011a)

Kondenzátorok

A kondenzátor a hűtőberendezésnek azon az egysége, amelyben a kompresszortól érkező hűtőközeg (gőz) a természetes hűtőközegnek (levegő, víz) leadja a hőmennyisége nagy részét, és újra folyadékká kondenzálódik. A hűtőközeg a kompresszorból mindig nagy nyomáson és magas hőmérsékleten, túlhevített állapotban érkezik a kondenzátorba. A kondenzáció befejezésével a hűtőközeg egészében folyadékká alakul vissza, és ezután sok esetben még valamilyen formában utóhűtést is kap, így hőmérséklete alacsonyabb lesz, mint a kondenzációs nyomáshoz tartozó telítési hőmérséklet. A kondenzátorokat a következő két csoportba soroljuk: vízhűtéses kondenzátorok és léghűtéses kondenzátorok. A léghűtéses kondenzátorok pedig lehetnek természetes- (csendes) vagy mesterséges légáramlásúak (Vasáros, 2020).

Közepes és nagy teljesítményekre általánosan elterjed a fekvő elrendezésű csököteges kondenzátorok alkalmazása, mely felépítése a 4. ábrán látható. A hőt felvevő víz a csövekben áramlik. A hűtőközeg a csövek és a köpeny közötti térben helyezkedik el. A hűtött csövek külső

felületén képződött kondenzátum onnan lecsorogva a köpeny aljára kerül és onnan távozik (Láng & Jakab, 1984).



4. ábra: Vízhűtésű, csőköteges kondenzátor (Láng & Jakab, 1984)

Elpárolgatók

Az elpárolgató egy felületi hőcserélő, amelyben az elpárolgó folyadékfázisú hűtőközeg hőt von el a hűtött közegtől, ez a berendezés azon része, mellyel az egész hűtési folyamat célját végrehajtjuk. Az elpárolgatóba, a hűtőközeg folyékony állapotba lép be és a hűtőtérből felvett hő hatására túlhevített gőz halmazállapotban lép ki.

Csoportosításuk az alapján történik, hogy milyen halmazállapotú közeget hűt az elpárolgató, mivel elsősorban ez szabja meg az elpárolgató konstrukcióját. Léghűtés esetén, száraz rendszerű elpárolgatókról beszélünk és a hűtött levegő áramlási módjától függően két csoportba osztjuk őket: természetes áramlású és mesterséges áramlásúakra. Kialakításuk ez alapján sok féle ként alakulhat. Ha folyadék a hűtött közeg, akkor jellemzően a hűtőközeg csőben áramlik, és a cső felületén keresztül veszi fel a körülötte lévő folyékony közegtől a hőt. Nagyon elterjedt kialakítás a levegőt hűtő ventilátoros elpárolgatók. A ventilátor használata számos előnyt nyújt elpárolgatók esetén, kiemelendő a levegőoldali hőátadási tényező javulása ventilátoros kialakítás esetén (Jakab, 2006).

Expanziós henger, adagoló

Az adagoló feladata a rajta keresztül áramló folyadék állapotú hűtőközeg nyomásának csökkentése a kondenzációs nyomásról az elpárolgási nyomásra. Szabályozza, hogy az elpárolgatóba jutó hűtőközeg tömege a hűtési teljesítménnyel arányos legyen.

Alkalmazható kapilláris cső, melynek feladata a folyadékállapotú hűtőközeg folyamatos táplálása és a közeg nyomásának csökkentése és a kompresszor leállítása esetén nyomáskiegyenlítés létrehozása a rendszerben. Előnye, hogy nincsenek mozgó alkatrészek,

olcsó és rendkívül üzembiztos a használata. Elterjedtebb adagoló eszköz a különböző szelepek (automatikus expanziós szelep, termosztatikus expanziós szelep), hiszen az elpárologtatóba áramló hűtőközege tömegét a szívónyomástól függően szabályozza és a berendezés hőmérsékletviszonyaitól függetlenül a szívónyomást automatikusan a kívánt állandó értéken tartja (Rácz, 2011a).

Ventilátorok

A hűtőkörfolyamat berendezésein kívül (kondenzátor, kompresszor, expanziós henger és elpárologtató) fontos említést tenni a hűteni kívánt térben való levegőmozgatásra szolgáló egységről is, vagyis a ventilátorról. Az 1. ábrán bemutatott légmozgatásos léghűtőberendezés alapvető egysége a ventilátor, melynek két fajtáját különböztetjük meg, a szívó oldali és a nyomó oldali ventilátort. Nyomó oldali ventilátor az, mely a hűtött levegőt a hűteni kívánt helyiségbe juttatja, és szívó oldali ventilátor, pedig az, amelyik a felmelegedett, terméktől hőt felvett levegőt elszívja a közegből és a hűtőkörfolyamatba visszajuttatja. A ventilátorok elhelyezése a termékek betárolási módjától és a hűtőkamra/ hűteni kívánt közeg kialakításától függenek.

Megkülönböztetünk axiális („propelleres”) ventilátorokat és radiális ventilátorokat. Axiális ventilátorok esetén a motor és a lapátok közös tengelyen kerülnek rögzítésre, valamint a tengely és a lapátok elfordulása a tengellyel párhuzamos légáramot hoz létre. Radiális ventilátor esetén a tengely irányban érkezik a szívóoldali levegő a lapáthoz, de sugárirányba fordulva halad át a lapátok között, ez miatt sokszor centrifugális ventilátornak is nevezik (Rácz, 2011b).

Mérőműszerek

A hűtőberendezések elemeit leszámítva kulcsfontosságú a jól megválasztott mérőműszer használata, hiszen a hőmérséklet, páratartalom és a rendszerben fellépő nyomásértékek mérése szükséges a hűtőberendezés üzemeltetéséhez. A különböző fizikai mennyiségek mérésére szolgáló mérőműszerek számos mérési elven működnek. Például a hőmérséklet mérésére szolgáló műszerek lehetnek folyadék tágulásán alapuló hőmérők, szilárd testek térfogatváltozásán alapuló hőmérők, sugárzáson alapuló hőmérsékletmérők vagy akár villamos mennyiségek mérésén alapuló hőmérők. A fontos viszont, hogy a mérőműszert jól válasszuk meg, a mérési tartománya elégséges legyen a körülményeknek, és ellássa a szükséges funkciókat (Fekete & Menyhárt, 1965).

Fekete és Menyhárt (1965) könyvében olvashatunk a légtechnikában alkalmazott mérőműszerek általános ismereteiről. A műszereket négy nagy fő csoportba sorolják. Az első

csoport a helyszínen jelző műszerek. A második csoportba a távmutatókat sorolják, melyek annyiban különböznek az előző csoportból, hogy az adó és a jelző külön helységben helyezkedik el. A harmadik csoport a szabályozó mérőműszerek, melyek kiegészülnek egy olyan mechanizmussal, amely képessé teszi a műszert egy folyamat elindítására, illetve megállítására. A negyedik csoportba tartoznak az író és számláló műszerek, melyek a mért értéket folyamatosan feljegyzik, például az idő függvényében, és ezekkel az értékekkel számolnak.

2.1.6. A hűtőberendezések automatizálásának kialakulása

A hűtőgépházak automatizálásának számos előnye van, mint a hatékonyság növelése, a működési költségek csökkentése, emberi erőforrás szükségének minimalizálása és a megbízhatóság javítása. Az automatizálás során különböző rendszerek és technológiák integrálása történik, amelyek lehetővé teszik a hűtőgép működésének valós idejű nyomon követését és irányítását. A PLC-k megjelenése a piacon ezt a fejlődést nagy mértékben növelte (erről a témáról 2.2. fejezetben bővebben beszélünk), viszont már a PLC-k megjelenése előtt törekedtek az automatizált hűtőberendezések kialakítására.

Gänger (1966) könyvében olvashatunk a PLC-k megjelenése előtti hűtőgépházak automatizálási módjáról és eszközeiről. Akkoriban számos általános módszert (feszültségmérő, hőmérséklet- és nyomásmérő stb.) és automatikaelemet (motorvédő, mágnesszelep, nyomás által vezérelt kapcsoló stb.) használtak, de kialakítottak számos speciális hűtőautomatika elemet is (önműködő adagolószelep, termosztatikus expanziósszelep, helyiségtermosztát stb.). Ezen eszközök fejlettebb változata mai napig részt vesznek a hűtőberendezések automatizálásában. Ezen kívül Gänger (1966) felsorolja az automatizálás feladatául kitűzött célokat, amiket a szabályozó- és kapcsolóberendezések végzik el. Ezek a célok a következők:

- Berendezés védelme rendellenes üzemelési körülmények folytán bekövetkező károsodás ellen.
- A hőmérsékletek állandó szinten tartása.
- A hűtőteltjesítmény hozzáigazítása a tényleges szükséglethez.
- A legkisebb hajtóigénnyel járó üzemelés beszabályozása.

Akkor már teljes automatizációról beszéltek, ha az előbb felsoroltak közül az első három funkciót ellátta a berendezés, valamint igen költségesnek számított a kialakítás, főleg mind a négy funkció esetén.

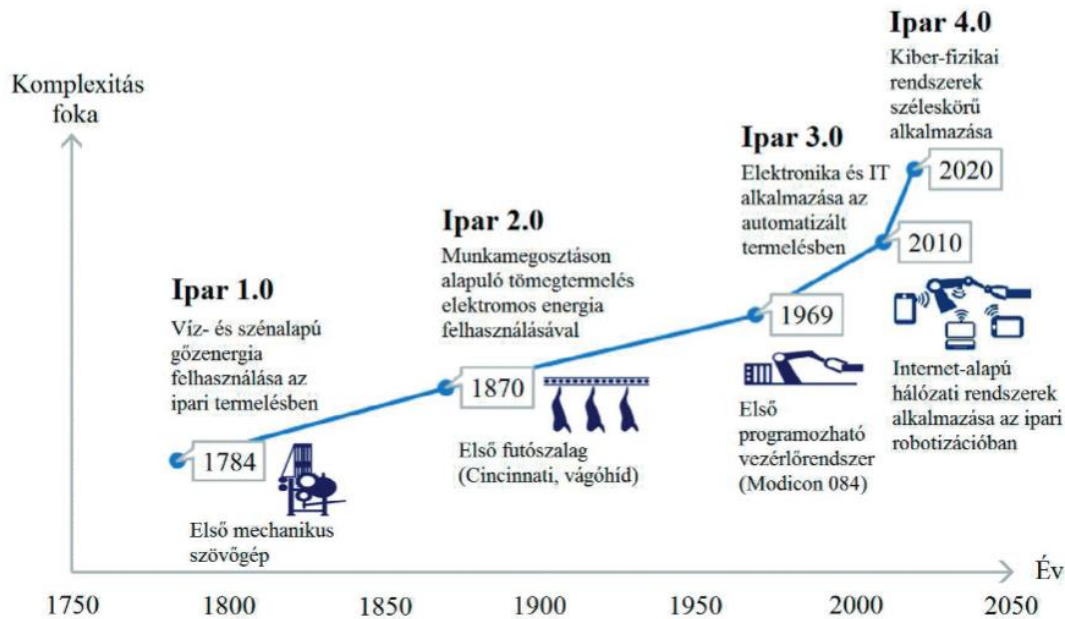
Thornhill és Lowrey (1993) esettanulmányában összefoglalják, egy hűtőraktár és érlelő üzem vezérlését. A tanulmányban említettek szerint alacsony költségű vezérlőrendszerek, amelyek intelligens programozható logikai vezérlőket (PLC-ket), személyi számítógépeket (PC), és PC-alapú SCADA (folyamatirányító és adatgyűjtő rendszer) rendszereket használtak, amely lehetővé tette a tárolók és érlelő területek környezeti feltételeinek (hőmérséklet, páratartalom, levegő nedvességtartalom) szoros ellenőrzését, így biztosítva a termékek frissességét és minimalizálva a pazarlást. A teljes hűtőraktárat SCADA rendszer felügyelte, melyen hűtési folyamatot egy PLC vezérelte a következő elemeken keresztül: párologtatók, kompresszorok, kondenzátorok, folyadékszivattyúk, ammónia detektálására szolgáló érzékelők. Ma már ezen funkciók alapvetőnek tekinthetőnek, ha automatizációról beszélünk és a technológia fejlődésének köszönhetően számos új funkcióval bővültek az automatizált hűtőberendezések.

2.2. Ipari automatizálás

Az ipari automatizálás célja, hogy bármilyen folyamatot nagyon kevés vagy emberi beavatkozás nélkül irányítson egy adott rendszer. Ez az egyik fő oka annak, hogy az iparágak egyre inkább az automatizált folyamatokat választják, mivel ezek rugalmas és költséghatékony működést biztosítanak jó versenyképességgel és magas hatékonysággal (Basant & Narendra, 2020).

Az utóbbi két-háromszáz évben három ipari forradalom volt és a negyedik most van kibontakozóban. Ezek mindegyikének megvoltak a maguk húzóerői és főbb sajátosságai, amelyek az egyes ipari forradalmak elkülönítésének is az alapját képezték. Az 5. ábra jól szemlélteti a különböző ipari forradalmak kiemelkedő eseményeit. A 3. ipari forradalom fő jellemzője az elektronika és IT alkalmazása az automatizált termelésben. A forradalom „kitörése” 1969-ra tehető, amikor is ez első programozható vezérlőrendszer, vagyis PLC megjelent. Jelenleg a 4. ipari forradalom idejében járunk, melyet 2010-ben kezdődött. Jellemzője a kiber-fizikai⁴ rendszerek alkalmazása, mint például az internet-alapú hálózati rendszerek alkalmazása az ipari robotizációban (Kiss & Tiner, 2021).

⁴ A kiber-fizikai rendszerek (Cyber-Physical Systems, CPS) olyan rendszerek, amelyekben a fizikai és digitális (kiber) komponensek szorosan együttműködnek. Például: érzékelők és beavatkozók valós idejű adatgyűjtése és digitális elemzése, majd ez alapján fizikai világ irányítása és optimalizálása.



5. ábra: Ipari forradalmak időrendben és fő húzóerejük (Kiss & Tiner, 2021)

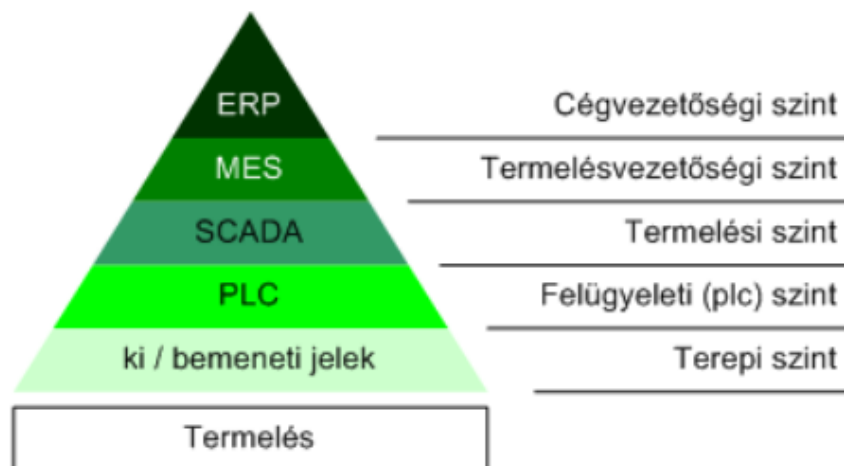
A PLC-eket széles körben alkalmazzák mai napig automatizált folyamatvezérlő rendszerekben, mivel egyszerűek, könnyen programozhatók, és időt takarítanak meg a relék, időzítők és más vezetékes vezérlők kiváltásával. Emellett megjelentek a 4. ipari forradalom legújabb automatizálási fejleményei, melyek számos előnyt kínálnak, mint például a folyamatok decentralizálása⁵, virtualizáció, interoperabilitás⁶ és valós idejű adatgyűjtés. Nincs egyetlen egységes meghatározása az Ipar 4.0-nak, de úgy értelmezhető, mint egy átfogó módszer, amely komplex gépeket, szenzorokat, aktuátorokat és szoftvereket kapcsol össze hálózatba a vállalati folyamatok előrejelzésére, szervezésére és az irányítási eredmények hasznosítására (Basant & Narendra, 2020).

2.2.1. Az automatizálási piramis

Az ipar 4.0 a különböző elemek hálózatokba kapcsolásán alapul. Az irányítórendszer szerepét intelligens kommunikációs rendszerek töltik be, melyek az automatizálási piramis (6. ábra) különböző szintjein mérnek meghatározott eredményeket.

⁵ A decentralizáció az irányítási vagy feldolgozási funkciók elosztása több, egymással együttműködő egységre vagy szintre, ahelyett, hogy egy központi vezérlőre támaszkodnánk.

⁶ Az interoperabilitás a rendszerek, eszközök és szoftverek közötti együttműködési képességet jelenti, függetlenül azok gyártójától vagy technológiájától.



6. ábra: Automatizálási piramis (Husi, 2016)

Az automatizálási piramis az ipari automatizálási rendszerek hierarchikus modellje, amely bemutatja, hogyan épülnek egymásra a különböző szintek az információáramlás és a gyártási folyamatok irányítása érdekében. A piramis egyes szintjei különböző funkciókat látnak el, a fizikai folyamatok vezérlésétől az üzleti szintű döntéshozatalig. Az automatizálási piramis segít strukturált módon megérteni, hogyan kapcsolódnak egymáshoz az ipari rendszerek különböző szintjei, lehetővé téve az adatok folyamatos áramlását és a folyamatok irányítását az érzékelőktől és beavatkozó szervektől egészen a stratégiai döntéshozatali felső vezető szintjéig (Boyer, 2009).

Az automatizálási piramis „lépcső fokai a következők:

- Terepi (ki/bemeneti jelek, Field level) szint: a legalsó szinten találhatók a fizikai eszközök, érzékelők, beavatkozó szervek. Ezek az eszközök közvetlen kapcsolatban állnak a gyártási folyamattal, adatokat gyűjtenek, valamint vezérlési parancsokat hajtanak végre. Példák eszközökre: helyszíni kijelzők, különféle mérőműszerek, hajtások, vészstopok, pneumatikus szelepek.
- Felügyeleti (PLC, Control level) szint: itt találhatók a vezérlőberendezések, melyek feldolgozzák a terepi szintről érkező adatokat, és a beavatkozó szervek működtetését végzik. Példák eszközökre: PLC (programozható logikai vezérlő), DCS (elosztott vezérlő rendszerek).
- Termelési (SCADA, Supervisory level) szint: Ezen a szinten találhatók a SCADA rendszerek (felügyeleti irányítási és adatgyűjtő rendszer), valamint a HMI-k (Human-Machine Interface). Ezek felelősek a valós idejű adatgyűjtésért, felügyelik a gyártási folyamatokat és lehetővé teszik az operátorok számára az ellenőrzést és a beavatkozást.

- Termelésvezetőségi (MES, Manufacturing Execution System- MES level) szint: Az MES-rendszerek felelősek a termelési folyamatok irányításáért és optimalizálásáért, a termelési adatok nyomon követéséért, valamint a minőségellenőrzésért.
- Cégvezetőségi (ERP, Enterprise Level) szint: Ez a legfelső szint, ahol az ERP (Enterprise Resource Planning) rendszerek működnek, amelyek a teljes vállalatirányítást szolgálják, beleértve az erőforrások tervezését, a készletgazdálkodást és a pénzügyi elemzéseket (Husi, 2016).

2.2.2. *PLC rendszerek*

A PLC-t (Programmable Logic Controller) arra tervezték, hogy megbízható módon vezérelje és monitorozza az ipari folyamatokat. A PLC-k képesek adatokat gyűjteni különféle érzékelőktől, feldolgozni azokat a programozott logika alapján, majd a megfelelő parancsokat küldeni a működtetőknél, például motoroknak, szelepeknek vagy lámpáknak.

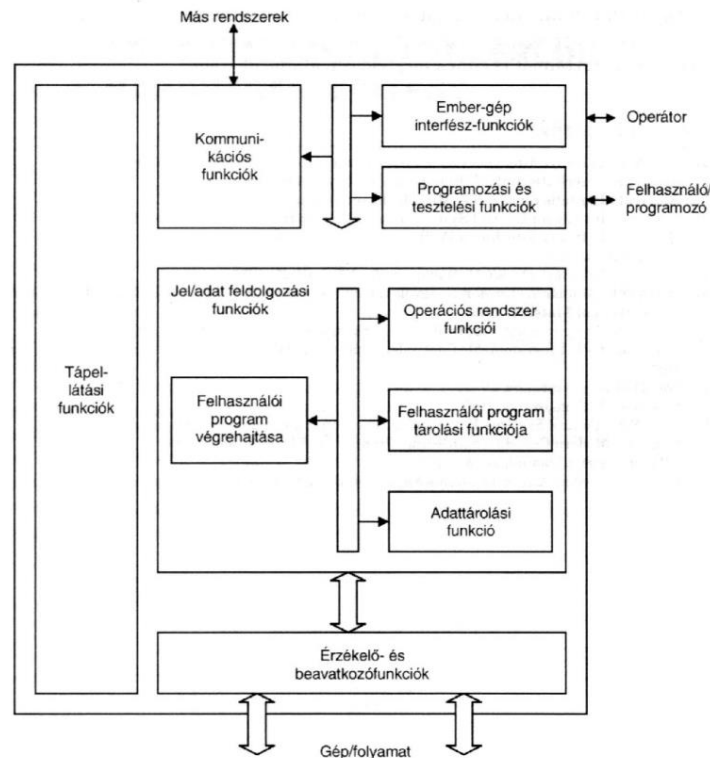
Az 1960-as évek végén fejlesztették ki ez első PLC-t, mely Modicon 084 néven vált ismerté. A cél az volt, hogy kiváltsák a hagyományos reléalapú vezérlőrendszereket, amelyek nehezen módosíthatóak voltak, valamint karbantartásuk időigényes volt. A PLC-k használata és fejlesztése egyszerűsítette és felgyorsította a gyártósorok újra programozását és karbantartását, ezen kívül hosszú élettartamúak, ipari kivitelben készülnek, képesek diagnosztikai feladatok ellátására, kommunikációs felületük révén hierarchikus, illetve osztott folyamatirányításra alkalmasak. Programozásuk számítógépes támogatással, grafikus szemléltetéssel történik, további szoftverek (SCADA, HMI) felhasználásával képes kielégíteni a folyamatirányítással szemben támasztott magasabb igényeket is (Ajtonyi, 2007).

A PLC-k kategorizálása ki/bemenetek száma, illetve mechanikai kiviteli módjuk szerint történik. Ki/be vonalak alapján megkülönböztetünk kis- (kevesebb mint 500 be/ki vonal), közepes- (500 és 5000 közötti be/ki vonal) és nagy (több mint 500 be/ki vonal) konfigurációjú PLC-eket. Mechanikai kivitel alapján pedig a következő csoportosítási módjuk ismert: kompakt PLC-k (hardverstruktúrájuk nem, vagy csak igen kis mértékben módosítható), micro-PLC-k (kompakt PLC speciális fajtája, amelyek tartalmazzák a HMI (ember-gép kapcsolat) szoftverfeltételeit is) és moduláris PLC-k (vezérlőberendezés, olyan modulokból építhető fel, mely modulok önmagukban ellátnak valamilyen speciális funkciót) (Csikós, 2014).

A PLC-k központi egysége a bemenetek és a kimenetek közötti, többnyire logikai kapcsolatokat időben egymás után és ciklikusan hajtja végre, a memóriába tárolt program alapján. A programozható logikai vezérlőkre vonatkozó IEC-61131 szabvány a következő funkciókkal definiálja PLC-eket:

- jel/adat feldolgozási funkció (signal/data processing)
- technológiai interfész funkció az érzékelők kezelésére, és a beavatkozók működtetésére
- kommunikációs funkció (PLC-PLC, PLC-számítógép, PLC-hálózat stb.)
- ember-gép interfész funkció (HMI)
- programozási, tesztelési, dokumentálási funkció
- tápellátási funkció (Hugh, 2007).

A szabvány szerinti funkciók sémája a 7. ábrán látható.



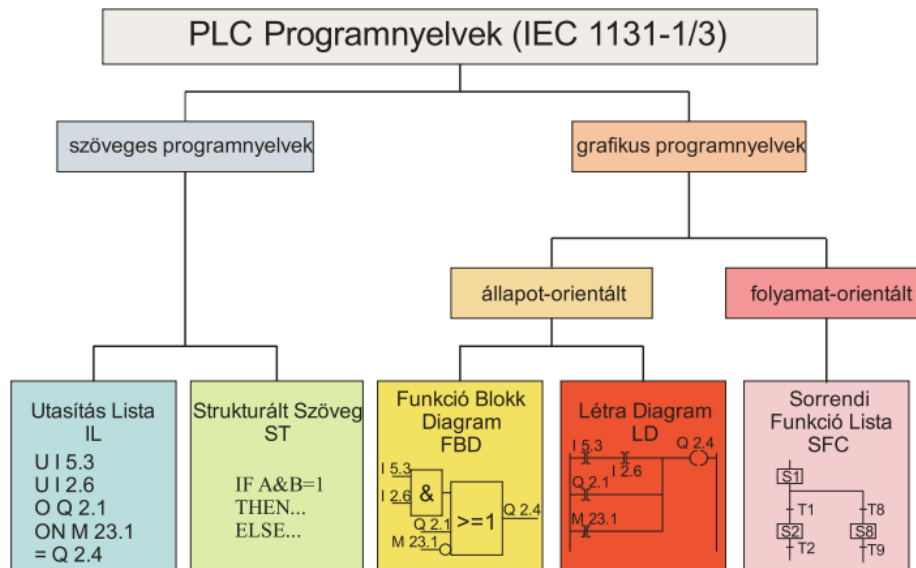
7. ábra: A PLC IEC-61131-es szabvány szerinti funkciói (Ajtonyi, 2007)

PLC-k szoftver rendszere és programozási nyelvei

A PLC-k hardvere univerzális, és önmagában nem, csak a felhasználói programmal együtt válik a konkrét irányítási feladatra alkalmassá. A PLC-k szoftvere két alap részre osztható, van az alapszoftver, mely állandó, és a felhasználói program, ami pedig a változó részt képviseli. Az alapszoftvert az operációs rendszer biztosítja, igen erősen gyártó- és típusfüggő, ezáltal különböző PLC-k esetén egyedi. A felhasználói programok, melyek a PLC-n futnak, speciális vezérléstechnikai és irányítástechnikai orientáltságú programnyelven írják.

A PLC-k fejlődése során számos programozási nyelvet fejlesztettek ki. Napjainkra a különböző felhasználói programnyelvek széles körét alkalmazzák, melyek egyes gyártói cégek eszközei közötti kompatibilitást lehetetlenné teszi, ezért nagy szükség volt, mint a felhasználók,

mint a gyártók részéről egy egységes nemzetközi szabványban rögzített felhasználói programnyelvek kifejlesztésére. Az IEC 61131-3 nemzetközi szabvány az egész világra kiterjedően egységesíti a felhasználói programnyelveket és ezek jelöléseit, ám fontos megjegyezni, hogy számos gyártó továbbra is alkalmaz a szabványtól eltérő programnyelveket. A programozási nyelvek felosztása, a 8. ábrán látható, minden programozási nyelvhez egy minta részlettel az adott programozási nyelv felépítéséről (Ajtonyi, 2007).



8. ábra: Szabványos PLC programnyelvek (Csikós, 2014)

A grafikus programozási nyelvek közül a LD (Ladder Diagram - Létra diagram) az egyik legismertebb PLC programozási nyelv. A kapcsolási rajzokhoz hasonló vizuális ábrázolást használ, amelyben az áramköri elemek egy létra fókain helyezkednek el. FBD (Function Block Diagram - Funkcióblokk diagram) egy olyan grafikus nyelv, amely a rendszer funkcióit különálló blokkokban ábrázolja. Az egyes blokkok különböző funkciókat valósítanak meg, és ezek bemeneti és kimeneti jeleken keresztül kapcsolódnak egymáshoz. SFC (Sequential Function Chart - Szekvenciális funkció diagram) pedig a folyamatok szekvenciális lépéseit mutatja. Az SFC nyelv lépésekből és átmeneteből áll, melyek a folyamat logikai sorrendjét mutatják. A szöveges programnyelvek közül az IL (Instruction List - Utasítás lista) a legrégebbi. Alacsony szintű szöveges nyelv, amely hasonlít az Assembly programozásra. Utasításokból áll, amelyeket soronként hajt végre, ellenben az ST (Structured Text - Strukturált szöveg) egy magas szintű, szöveges programozási nyelv, amely a Pascalhoz és a C nyelvhez hasonlít. Az ST programok bonyolult algoritmusokat és feltételes utasításokat tartalmazhatnak (Hugh, 2007).

2.2.3. SCADA rendszerek

A SCADA (Supervisory Control and Data Acquisition) rendszerek egy olyan ipari automatizálási technológiai platformot jelentenek, amelyek lehetővé teszik a folyamatok és berendezések valós idejű megfigyelését, ellenőrzését és adatgyűjtését távoli helyszíneken. A SCADA rendszerek folyamatosan adatokat gyűjtenek a folyamatokról és berendezésekről, feldolgozzák ezeket az adatokat, és az eredményeket a kezelők számára megjelenítik. Ha hiba esetén riasztást küldenek az operátoroknak, akik szükség esetén beavatkozhatnak, vagy akár automatikus korrekciós műveleteket is végrehajthatnak. Ezek alapján a SCADA rendszerek fő funkciói a következők:

- Valós idejű felügyelet: gyors reagálás meghibásodások és rendellenességek esetén, ezzel optimalizálva a folyamatot és növelve a hatékonyságot.
- Nagy mennyiségű adat tárolása és elemzése: teljesítmény optimalizálására és a karbantartási igények előrejelzésére, valamint a mért adatok visszatekintésére.
- Valós idejű megjelenítés: az operátorok nyomon követhetik a folyamatok állapotát.
- Automatikus és kézi vezérlés: a rendszer automatikus beavatkozásokra is képes a beállított paraméterek alapján, de lehetőséget biztosít a kézi beavatkozásra is.
- Központosított irányítás: egy helyről felügyelhető és vezérelhető az egész rendszer, akár több telephely is.
- Riasztások és hibaértesítések: a rendszerek figyelmeztetéseket küldenek az operátoroknak, amikor a paraméterek túlmutatnak a beállított határértékeken, ami lehetővé teszi a gyors beavatkozást és a károk minimalizálását.
- Karbantartás támogatása: az előzményadatok elemzésével lehetőség nyílik a megelőző karbantartásra, amely meghosszabbítja az eszközök élettartamát és csökkenti az állásidőt (Boyer, 2009).

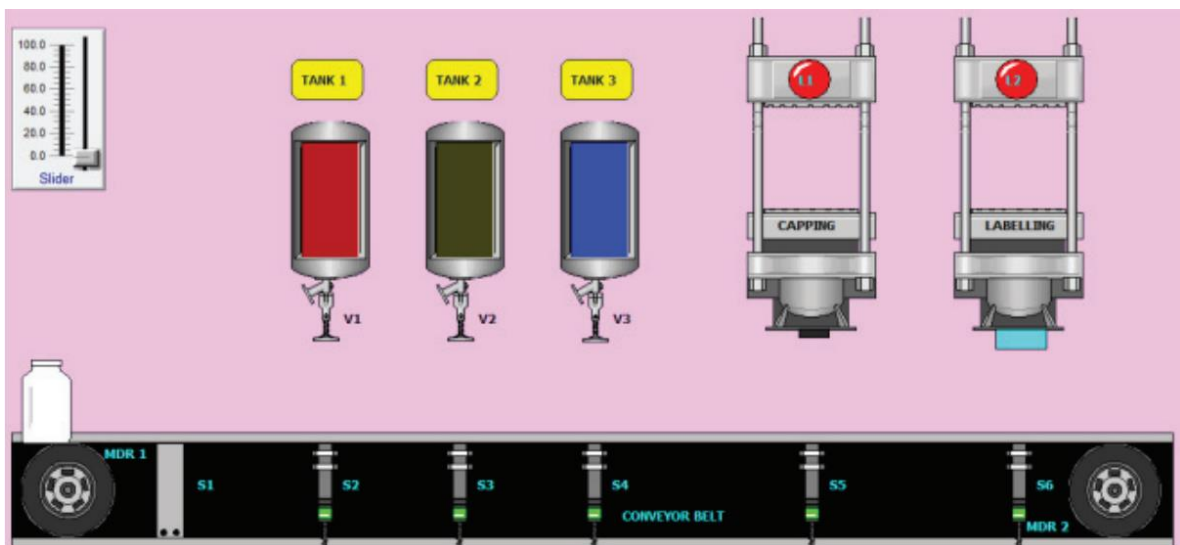
A SCADA rendszerek több kulcsfontosságú elemre épülnek, és ezen elemek együttesen biztosítják a folyamatok felügyeletét és irányítását. Az előző fejezetben tárgyalt PLC-k és más programozható vezérlésre vagy adatgyűjtésre szolgáló berendezéseken kívül fontos szerepet játszanak a folyamat irányításában a HMI-k. A HMI egy grafikus felhasználói felület, melyen keresztül tudják felügyelni az operátorok a folyamatot, és szükség esetén beavatkozhatnak. Az adatgyűjtő és feldolgozó szerverek, valamint adatbázisok fogadják az adatokat, tárolják és feldolgozzák azokat. Képesek az adatok naplózására, előzmények kezelésére és statisztikai elemzések készítésére is. Mind ezen felsorolt elemek, eszközök, szerverek összekötéséért felelősek a kommunikációs hálózatok, például Ethernet, vagy a vezeték nélküli hálózatok. A

kommunikációs protokollok, mint például a Modbus biztosítják a megbízható adatátvitelt az eszközök között (Ajtonyi, 2008).

Esettanulmány egy PLC és SCADA tervezési és megvalósítási folyamatáról

Basant és Narendra (2020) tanulmányában egy automatizált palacktöltési folyamat megjelenítésére szolgáló SCADA (Wonderware Intouch SCADA szoftver) rendszer, és a folyamat vezérlésére szolgáló PLC (OMRON- NX1P2-9024DT1) program megtervezéséről, elkészítéséről és teszteléséről olvashatunk. A tanulmány felhívja a figyelmet arra, hogy első és legfontosabb lépés az adott technológia megismerése és a teljes folyamat átlátható részekre tagolása, a további munka elvégzésének megkönnyítése érdekében. A következő fontos lépés a ki- és bemenetek (szenzorok és aktuátorok), valamint az egyéb komponensek felmérése és megismerése, és ezen elemek PLC és SCADA rendszerrel való kommunikációs hálózat megtervezése és kiépítése. A programozás és folyamatok megvalósítása előtt kiemelik az esettanulmányban, hogy fontos még felmérni a SCADA-val szemben támasztott elvárásokat, a felhasználók által, mint például a riasztások, és hogy milyen adatok jelenjenek meg a HMI-n az operátorok számára. Az elkészült PLC program és SCADA rendszer tesztelésére, első körben pedig szimulációt ajánlanak.

A tanulmányban bemutatott szimuláció elkészülésének célja a hatékonyság növelése és az emberi beavatkozás minimalizálása a palackok különböző szakaszokon történő áthaladásával. A tanulmány szerint a rendszer sikeresen bevezethető, csökkentve a munkaerőigényt, növelve a termelékenységet és javítva a termékminőséget. Az említett szimuláció folyamatmodellje a 9. ábrán látható, ahol is az úgynevezett Wonderware Intouch SCADA szoftvert használták.



9. ábra: Folyamatmodell a SCADA Wonderware Intouch szoftverben (Basant & Narendra, 2020)

3. ANYAGOK ÉS MÓDSZEREK

Ebben a fejezetben ismertetem azon hardveres és szoftveres eszközöket, melyek a SCADA rendszer elkészítésében szerepet játszottak valamilyen módon. Ismertetni fogom a hűtőberendezés és a hűtőgépház azon eszközeit, melyekből információt nyertem, a szoftvert, melyben a SCADA rendszert készítettem, és azon egyéb információ forrásokat, melyek rendelkezésemre álltak a munka megvalósítása során.

3.1. Hardveres eszközök

A hűtőgépház vezérlését egy moduláris Omron PLC hajtja végre, melyet egy kommunikációs kártyával bővítettünk, hogy a hálózatra csatlakoztassuk, és adatokat tudjunk kinyerni belőle (a PLC-n futó program, mely vezérli a hűtést).

3.1.1. Omron PLC

A PLC meglévő és új modulját a 1. táblázatban fogom ismertetni, rövid leírással. Az új modult zöld színnel fogom jelölni, mely beépítésre került.

1. táblázat: Omron PLC felépítése

Típus	Mennyiség	Leírás
CJ1M-CPU11	1	Kompakt, moduláris PLC, amelyet kis és közepes méretű automatizálási rendszerekhez fejlesztettek ki
CJ1W-ID211	4	Digitális bemeneti modul, amelyet kifejezetten 24V DC jelű bemeneti jelek fogadására terveztek
CJ1W-OC211	1	Digitális kimeneti modul, amelyet kifejezetten 24V DC jelű kimeneti jelek küldésére terveztek
CJ1W-OD211	2	Digitális kimeneti modul, amelyet 12-24V DC jelű kimeneti jelek küldésére terveztek
CJ1W-DRM21	1	Lehetővé teszi, hogy a PLC csatlakozzon a DeviceNet hálózathoz, és adatokat cseréljen különféle eszközökkel
CJ1W-PA202	1	Tápegységmodul, amely biztosítja a szükséges tápellátást a PLC CPU egységének és a hozzá csatlakoztatott I/O moduloknak
DRT2-AD04	1	Analóg bemeneti modul, amely DeviceNet hálózatra csatlakozik, és lehetővé teszi analóg jelek fogadását különböző ipari érzékelőktől
CJ1W-ETN21	1	Lehetővé teszi a PLC hálózati csatlakoztatását és Ethernet hálózaton keresztüli kommunikációt

3.1.2. Érzékelők

A különböző érzékelők és beavatkozó szervek a PLC I/O moduljaihoz csatlakoztatva biztosítottak számomra információt, viszont a nyomásmérő és a hőmérsékletmérő műszerek pontos típusának ismeretére szükségem volt, a tőlük érkező adatot skálázni kellett a megfelelő módon.

A nyomásmérőről megszerzett információk: nyomásmérő belső membránnal, rozsdamentes acélból, alsó csatlakozási kivitelben, méréshatár 0-tól 1000 bar-ig. Kimeneti jel 4...20mA. Gyártó: SUKU Druck & Temperature Típusa: 3340.

A hőmérsékletmérőről megszerzett információk: Ellenállás-hőmérő, rozsdamentes acélból, alsó csatlakozási kivitelben, méréshatár: -50 °C-tól +400 0°C-ig. Kimeneti jel 4...20mA. Gyártó: SUKU Druck & Temperature Típusa: 8010.

3.2. SCADA szoftver

SCADA tervező szoftver mely rendelkezésemre állt a cég által meghatározott, Ignition SCADA szoftver volt. Az Ignition vizualizációs eszközeivel kombinálva lehetővé teszi, hogy a felhasználók hatékony és reszponzív HMI és SCADA alkalmazásokat hozzanak létre, amelyekkel ellenőrizhetik, vezérelhetik és archiválhatják ipari folyamataik adatait. Az Inductive Automation adta ki 2010 januárjában.

Webalapú, és webes technológiák segítségével telepíthető és futtatható. Az Ignition Platform kezelése, vezérlése és konfigurálása egy gyors és könnyen használható webes felületen keresztül történik. A webalapú kommunikációs technológiáknak köszönhetően az Ignition képes PLC-khez, OPC szerverekhez csatlakozni, valamint SQL adatbázisokat elérni bármilyen platformon. Az Ignition SCADA modulok olyan funkciókat nyújtanak, mint: valós idejű állapotellenőrzés (továbbiakban: Real-Time Status Control), riasztás (továbbiakban: Alarming), jelentéskészítés (továbbiakban: Reporting), adatgyűjtés (továbbiakban: Data Acquisition), szkriptek (továbbiakban: Scripting), ütemezés (továbbiakban: Scheduling).

Az Ignition projektek mobil eszközökön vagy érintőképernyős panelekből indíthatók bárhol, és bármely PC-ről elindíthatók, függetlenül az operációs rendszertől.

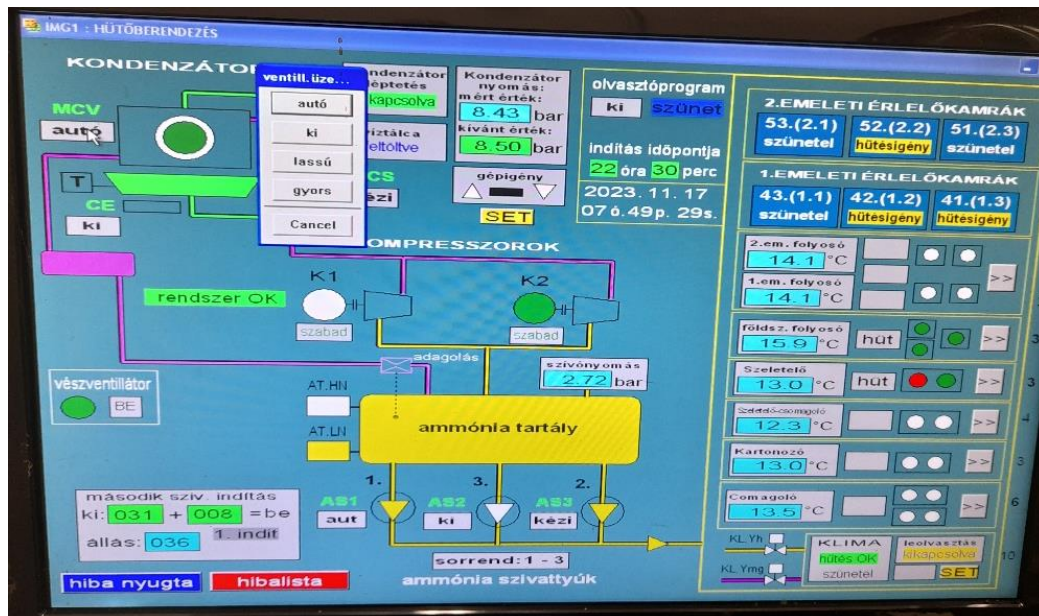
A SCADA készítés során a szoftver felhasználói kézikönyve és a gyártó által készített oktatóvideók álltak rendelkezésemre. Az Ignition szoftver felhasználói kézikönyve és az Inductive University, melyen elérhetőek a szoftver használatához a gyártó által készített oktatóvideók, az alábbi linken érhetők el: <https://www.docs.inductiveautomation.com/>.

3.3. Egyéb információ források

A hűtőgépházat vezérlő PLC hálózatba kötése előtt kimentettük a PLC-n futó programot, valamint a régi SCADA rendszerből a tag listát. Ezen kívül a régi SCADA rendszerről készített fotók és a rendszer villamos kapcsolási rajza állt rendelkezésemre.

3.3.1. Meglévő SCADA felület

Az telephelyen lévő kollegák lefényképeztek 20 darab képernyő fotót (10. ábra) a régi SCADA rendszer kinézetéről és funkcióiról, valamint egy Teams-es megbeszélés keretein belül segítségemre voltak a rendszer mögötti technológia megértésében, és készségesen átbeszéltük a meglévő funkciókat, melyeket használnak a kezelőfelületen. Ezen felül kértek pár módosítást, valamint új funkciót, mint például az adatok kimentése Excel táblázatba és Pdf-be grafikon formátumban, a kiválasztott időintervallum szerint.



10. ábra: Régi SCADA rendszer képernyő fotó

3.3.2. Wizcon tag lista

A régi SCADA rendszer tageinek listáját kimentettük egy Excel táblázatba, melyben összesen 220 sor szerepel, mindegyik sorban egy tag volt feltüntetve. Számomra a dokumentum hasznos adatokat szolgáltatott a rendszer működésének visszafejtésében, és a PLC-n futó program megértésében. Valamint abban, hogy a PLC melyik I/O, vagy memória címén milyen érték van. A 11. ábrán látható a kimentett tag lista, piros négyzettel emeltem ki a számomra hasznos információkat, melyeket felhasználtam. Az Excel táblázat B oszlopában a tag rövidített megnevezése szerepel, míg a C oszlopban egy leírás. Az S oszlopban pedig a PLC megcímezésére utaló információ.

ID	Name	Description	UserGroups	HistoryRecordRe	DdeType	DdeUpdate	Ddi	DdeBlockCol	Ownerdata	SourceTyp	UpdateMc	FormatTyp	VPINumber	VPIAddress
77	T1_UMK	Készárú raktár hűtés üzemmódkapcsoló	1	0	0	0	0	0	0	1	0	0	0	00H00
78	T2_UMK	Expediáló hűtés üzemmódkapcsoló	1	0	0	0	0	0	0	1	0	0	0	00H01
79	T3_UMK	Kartonozó hűtés üzemmódkapcsoló	1	0	0	0	0	0	0	1	0	0	0	00H02
80	T4_UMK	Szeletelő hűtés üzemmódkapcsoló	1	0	0	0	0	0	0	1	0	0	0	00H03
81	T51_UMK	1.emeleti folyosó hűtés üzemmódkapcsoló	1	0	0	0	0	0	0	1	0	0	0	00H04
82	T52_UMK	2.emeleti folyosó hűtés üzemmódkapcsoló	1	0	0	0	0	0	0	1	0	0	0	00H05
83	T6_UMK	Földszint folyosó hűtés üzemmódkapcsoló	1	0	0	0	0	0	0	1	0	0	0	00H06
84	KLIMA_UMK	Klíma olvasztás engedélyezés kapcsoló	1	0	0	0	0	0	0	1	0	6	0	00h2003
85	LP_TA	szívónyomás	1	300000	0	0	0	0	0	1	0	0	0	160
86	T1_HIGIL	Készárú raktár hűtésigény jelzés	1	0	0	0	0	0	0	1	0	6	0	00i02000
87	T2_HIGIL	expediáló hűtésigény jelzés	1	0	0	0	0	0	0	1	0	6	0	00i02001
88	T3_HIGIL	kartonozó hűtésigény jelzés	1	0	0	0	0	0	0	1	0	6	0	00i02002
89	T4_HIGIL	szeletelő hűtésigény jelzés	1	0	0	0	0	0	0	1	0	6	0	00i02003
65	KLIMA_HIG	Klíma hűtésigény jelzés	1	0	0	0	0	0	0	1	0	6	0	00i002404
66	KLIMA_OK	Klíma felé OK jel kiadása	1	0	0	0	0	0	0	1	0	6	0	00i002403
67	KLIMA_OLV	Klíma felé olvasztás jel kiadása	1	0	0	0	0	0	0	1	0	6	0	00i002406
68	T1_MÉRT	T1. Készárú raktár mért hőmérséklet	1	300000	0	0	0	0	0	1	0	1	0	50
69	T2_MÉRT	T2. Expediáló mért hőmérséklet	1	300000	0	0	0	0	0	1	0	1	0	56
70	T3_MÉRT	T3. Kartonozó mért hőmérséklet	1	300000	0	0	0	0	0	1	0	1	0	62
71	T4_MÉRT	T4. Szeletelő mért hőmérséklet	1	300000	0	0	0	0	0	1	0	1	0	68

11. ábra: Részlet a wizcon tag listából

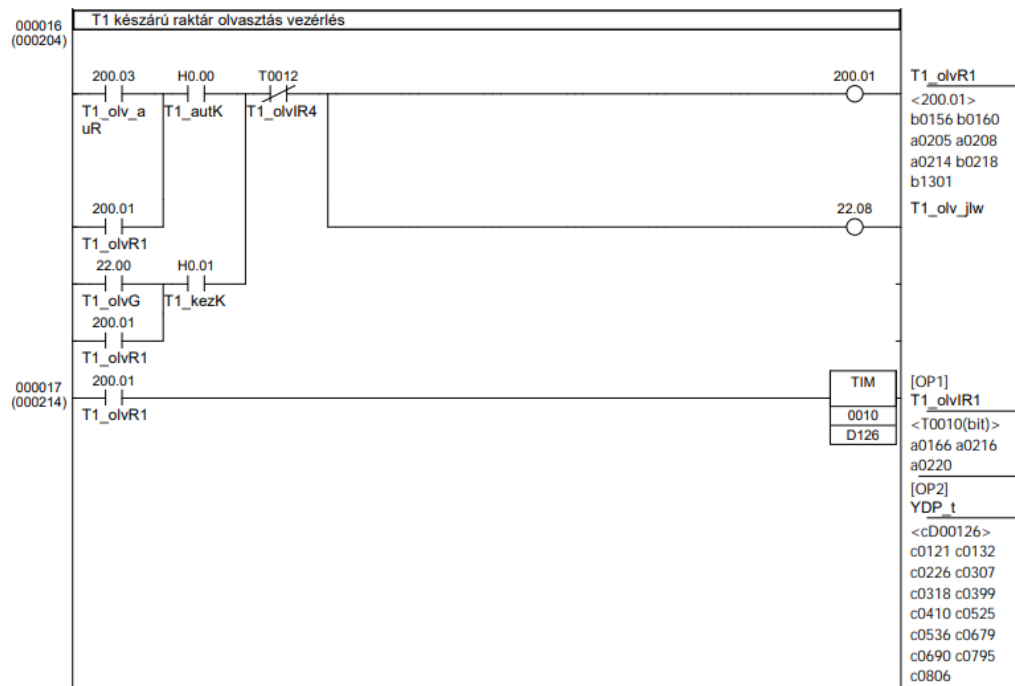
Ezen kívül a régi SCADA rendszerből kimentettük az alarmok listáját egy 33 sorból álló Excel táblázatba. Mindegyik sor egy alarmot tüntetett fel, melyről egy részlet a 12. ábrán látható. A képen piros négyzettel kiemelttem a számomra hasznos információkat. Az Excel táblázat B oszlopában látható az a feltétel, ami esetén hibajelzést kell generálni, és a I oszlopban, pedig egy leírás a hibajelről.

A	B	C	D	E	F	G	H	I	J
ID	Expression Family	Severity	Class	Zone	Attributes	InfoFile	Text	Attributes:	
0	@m11_mjl>1	0	1,61E+09	0	18		Készárú raktár 1.ventillátor hiba	0	
1	@m12_mjl>1	0	1,61E+09	0	18		Készárú raktár 2.ventillátor hiba	0	
2	@m21_mjl>1	0	1,61E+09	0	18		Expediáló 1.ventillátor hiba	0	
3	@m22_mjl>1	0	1,61E+09	0	18		Expediáló 2.ventillátor hiba	0	
4	@m31_mjl>1	0	1,61E+09	0	18		Kartonozó 1.ventillátor hiba	0	
5	@m32_mjl>1	0	1,61E+09	0	18		Kartonozó 2.ventillátor hiba	0	
6	@m411_mjl>1	0	1,61E+09	0	18		Szeletelő 1.léghűtő 1.ventillátor hiba	0	
7	@m412_mjl>1	0	1,61E+09	0	18		Szeletelő 1.léghűtő 2.ventillátor hiba	0	
8	@m421_mjl>1	0	1,61E+09	0	18		Szeletelő 2.léghűtő 1.ventillátor hiba	0	
9	@m422_mjl>1	0	1,61E+09	0	18		Szeletelő 2.léghűtő 2.ventillátor hiba	0	
10	@m511_mjl>1	0	1,61E+09	0	18		1. emelet folyosó 1. léghűtő ventillátor hiba	0	
11	@m512_mjl>1	0	1,61E+09	0	18		1. emelet folyosó 2. léghűtő ventillátor hiba	0	
12	@m521_mjl>1	0	1,61E+09	0	18		2. emelet folyosó 1. léghűtő ventillátor hiba	0	
13	@m522_mjl>1	0	1,61E+09	0	18		2. emelet folyosó 2. léghűtő ventillátor hiba	0	
14	@m61_mjl>1	0	1,61E+09	0	18		földszint folyosó 1. léghűtő ventillátor hiba	0	
15	@m62_mjl>1	0	1,61E+09	0	18		földszint folyosó 2. léghűtő ventillátor hiba	0	
16	@m63_mjl>1	0	1,61E+09	0	18		földszint folyosó 3. léghűtő ventillátor hiba	0	
17	@AS1_jar>1	0	1,61E+09	0	18		1. ammóniaszivattyú hiba	0	

12. ábra: Részlet a wizcon alarm listából

3.3.3. Omron PLC program

A PLC-n futó programot kimentettük, így a programkód és a taglistának köszönhetően vissza tudtam fejteni a PLC-n futó programot, és ezáltal megérteni a rendszer működését. 59 darab A4-es formátumba kimentett létradiagramm nyelven íródott PLC program egyik részlete a 13. ábrán látható.



A kimentett Pdf-ek a következő részegységekre bonthatóak:

- Alap program - 6 oldal (indítás, nyugta gombok egyesítése, ammóniatartály vészszintjelzések, kompresszor állapotjelzések, nyomásértékek, vészventilátor állapotjelzések, klíma állapot jelzések, klíma olvasztóprogram vezérlése)
- Ammónia szivattyúk - 9 oldal (ammónia szivattyúk kiválasztott üzemmódjainak megfelelő működés, állapotjelzések, hűtés szükséglet számítása és szivattyú igényszám, sorrendkapcsoló által szivattyú prioritizálási program)
- Hibajelek – 4 oldal (gyűjtött és végül kijelzett hibajelek, hibalámpák vezérlése)
- Kondenzátor – 8 oldal (ventilátor-, szivattyú-, és vízfűtés üzemmód, valamint állapotjelzések, vízhiány és tartós szinthiány jelzések, víztálca állapotjelzések, semlegeszóna számítás, kondenzátor léptetés és gépigény számítás)
- Olvasztóprogram – 5 oldal (olvasztóprogram üzemmód és állapotjelzés, szekciók és klíma olvasztóprogram indítás jeleinek generálása)
- T1 készárú raktár – 4 oldal (készárú raktár hűtésigény számítása, hűtés vezérlése, olvasztóprogram vezérlése, motor és szelep vezérlés és állapotjelek)
- T2 expedáló üzem - 4 oldal (expediáló üzem hűtésigény számítása, hűtés vezérlése, olvasztóprogram vezérlése, motor és szelep vezérlés és állapotjelek)
- T3 kartonozó üzem - 4 oldal (kartonozó üzem hűtésigény számítása, hűtés vezérlése, olvasztóprogram vezérlése, motor és szelep vezérlés és állapotjelek)

- T4 szeletelő üzem - 5 oldal (szeletelő üzem hűtésigény számítása, hűtés vezérlése, olvasztóprogram vezérlése, motor és szelep vezérlés és állapotjelek)
- T5 emeleti folyosók - 6 oldal (emeleti folyosók hűtésigény számítása, hűtés vezérlése, olvasztóprogram vezérlése, motor és szelep vezérlés és állapotjelek)
- T6 földszinti folyosó - 4 oldal (földszinti folyosó hűtésigény számítása, hűtés vezérlése, olvasztóprogram vezérlése, motor és szelep vezérlés és állapotjelek)

3.3.4. Vezérlő szekrény kapcsolási rajza

A vezérlő szekrény kapcsolási rajzai is segítségül szolgáltak számomra. 155 darab A4-es formátumú kapcsolási rajz állt rendelkezésemre a munka során. Lényegesebben kevesebb információt szolgáltatott nekem ez a forrás, mind a képernyőképek, a tag és alarm listák, valamint a PLC programok, viszont nem számít teljesen elhanyagolhatónak sem, információk szempontjából.

4. ÖNÁLLÓ MUNKAVÉGZÉS

Ebben a fejezetben ismertetem a munkám fő részeit, a programok visszafejtését és a tagek kezelését, valamint magát a megvalósítást. A megvalósítás egy hosszas folyamat volt, sok teszteléssel, egy erre a célra szánt tesztszerveren, és az ebből adódó módosításokkal. A fejezetben bemutatott főbb megvalósítások, a végleges program kódokat és beállításokat fogják tükrözni.

4.1. Célkitűzés megismétlése, elvárások ismertetése

Egy magyarországi élelmiszeripari cég telephelyén lévő hűtőgépház vezérlése egy Omron PLC-n, és egy hozzá tartozó SCADA rendszeren futott. A SCADA régi és elavultnak számított, valamint csak és kizárólag Windows XP-s operációs rendszerű számítógépre telepíthető. A cég a Windows XP operációs rendszert már nem támogatta, így a hálózatra sem lehetett felcsatlakoztatni a számítógépet, hogy távolról is elérje a kezelő. A SCADA rendszer felhasználója csak akkor tudott módosítást végezni, és értesülni a hűtőgépház állapotáról, ha fizikálisan is a számítógép közelében volt. Esetleges meghibásodás esetén a SCADA nem telepíthető újra, hiszen a szükséges operációs rendszer már nem támogatott a cégcsoporton belül.

Ezen indokok miatt, szükségszerűnek láttam a SCADA rendszer korszerűsítését. A fejlesztés során a rendszert üzemeltető felhasználók további új igényei is beépítésre kerültek a rendszerbe. Ez lehetővé tette, hogy a felhasználók számára a rendszer a lehető leghatékonyabb felhasználást nyújtsa.

Az új SCADA rendszerrel szemben támasztott elvárások:

- A cégcsoport fő hálózatára csatlakozott eszközről elérhető legyen, telephelytől és eszköztől független.
- A régi SCADA kinézete és felépítése alapvetően maradjon meg, a rendszer felhasználóinak megszokása miatt.
- Azon részek kivonása a SCADA rendszerből, melyek jelenleg nem üzemlenek és terv szerint nem is fognak (például a T2 expedáló üzem), valamint meglévő üzemrészek átnevezése a jelenleg aktuális megnevezésük szerint.
- Alarm rendszer kiépítése, és e-mail értesítés küldése hiba esetén az illetékes felhasználóknak.
- Felhasználói szintek kezelése, és felhasználói szinttől függően elérhető funkciók (mért értékek, grafikonok megtekintésére és exportálása bejelentkezés nélkül legyen lehetőség,

viszont nyugtázás és alarm lista törlésére és bármilyen érték átírására csak a megfelelő autentikációval legyen lehetséges).

- Mért értékek tárolása 3 évre visszamenően, és grafikon formájában megjelenítés időintervallum kiválasztása alapján.

- Értékek időintervallum szerinti exportálása, Excel táblázat és Pdf formátumba grafikonon.
- Munkám célja a felsorolt elvárások teljesítése a felhasználók és a vezetőség számára, valamint a cégben használt SCADA készítési standardok betartása (használt megnevezések, sémák stb.), hogy a későbbiekben könnyedén módosítható és átlátható legyen más cégben dolgozó kollégák számára.

4.2. PLC-n futó programrészek folyamatábrája

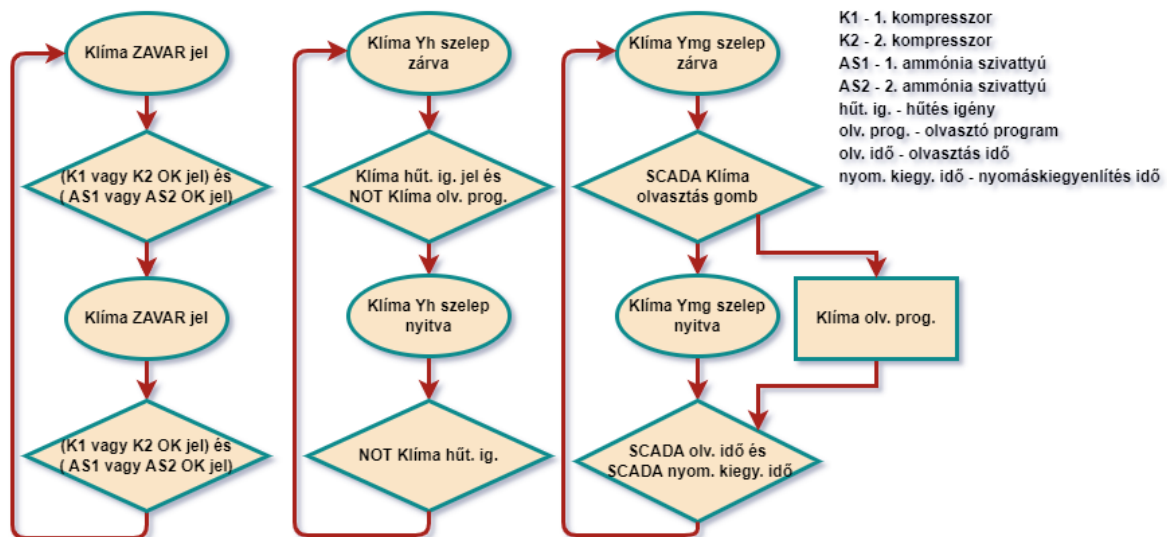
A 3.3.3. fejezetben tárgyalt PLC program részegységeinek megértésében nagy szerepet játszottak a 3.3.2 fejezetben tárgyalt tag listák, melyekben rövid leírást olvashattam az adott értékekről, és néhol a rövidített megnevezés megegyezett, vagy nagy mértékben hasonlított a tag listában szereplő rövid megnevezésre.

A létra diagram nyelven íródott PLC programokról készítettem néhány egyszerűsített folyamatábrát, a program működésének megértése érdekében. A 3.3.3. fejezetben tárgyalt részegységek megnevezése alapján ismertetem ezen program részletek működését. A programok visszafejtéséhez az Omron PLC felhasználói kézikönyvét használtam, mely az alábbi [linken](https://assets.omron.eu/downloads/manual/en/w451_cp1_cpu_unit_programming_manual_en.pdf) érhető el: https://assets.omron.eu/downloads/manual/en/w451_cp1_cpu_unit_programming_manual_en.pdf.

4.2.1. Alap program

Az alap program első felében a külső fizikális indító jel fogadása, és az üzemkész állapot külső jel kiküldése szerepel. Ezen kívül a külső fizikai nyugta gombot, és a SCADA-ban szereplő nyugta gombot egyesíti. Az alapprogramból derült még ki számomra, hogy az ammónia tartály vészszint jelzéseit egy-egy belső memória változóban tárolja. Az ammónia tartály felső vészszint jelzésére szolgáló memória érték, ha nagyobb mint 1, akkor hibát jelez, és ha az értéke 2 akkor tartós hibát jelent, ez ugyanígy igaz az ammónia tartály alsó vészszint jelzésére is. A két darab kompresszor állapotjelzései is az alap programban vannak elmentve, valamint a SCADA-ban lévő kompresszor szabad, illetve tiltó kapcsolójával lehet kiküldeni a leállás és indítás jelet. A kompresszorok ezen felül külső jelekből az alábbi állapotokat veheti fel: JÁR (a kompresszor üzemel), OK (a kompresszor rendben működik), WARM (a

kompresszor hőfoka nem megfelelő). A kompresszor rendszerfeltétele jelzés, az ammónia tartály szintjétől függően változik, és egy változóban menti el a jelét. Az alap programban megjelenik még 2 érték átadása memóriába, az ammónia tartálynál fellépő szívónyomás és a kondenzátornyomás. A programban ezen kívül az úgynevezett vészventilátort lehet bekapcsolni a SCADA-ban lévő kapcsolóval, valamint 2 külső jelzést kapunk a ventilátortól, az egyik a JÁR (ventilátor üzemel), a másik az ARH20% (alsó robbanási határ, a legkisebb koncentráció, ahol az adott anyag már a normál légköri viszonyok mellett robbanni képes). Az alap program utolsó része pedig a klíma működésére vonatkozott, melynek leegyszerűsített folyamatábráját a 14. ábrán mutatom be. A Klíma programrészében egy Klíma OK jel mellett a Yh (hűtő folyamathoz tartozó szelep) és a Ymg (meleggáz szelep, az olvasztáshoz tartozó szelep) vezérlése történik. A SCADA-ból adott klíma olvasztás kapcsoló, és a hozzá tartozó olvasztó idő és nyomáskiegyenlítési idő által.

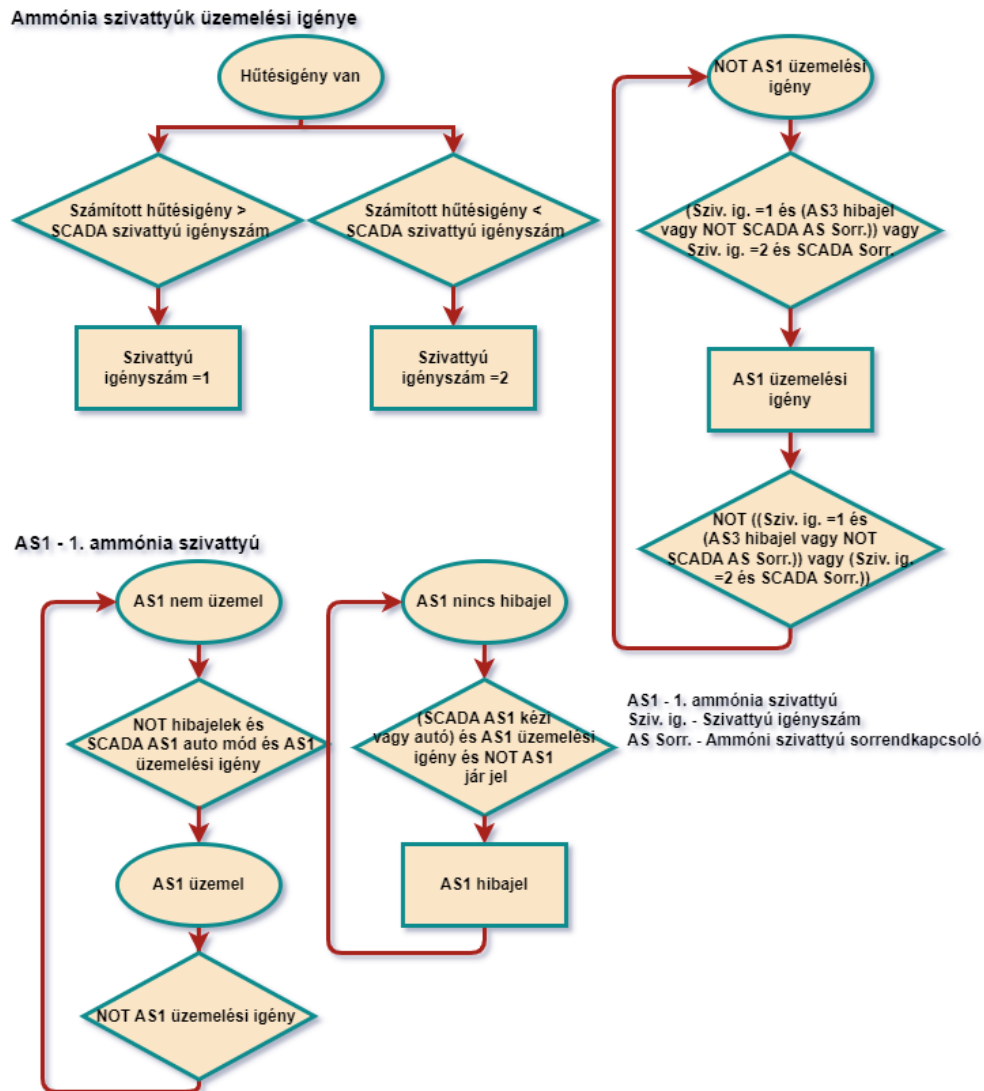


14. ábra: Klíma működésének folyamatábrája

4.2.2. Ammónia szivattyúk programja

Az ammónia szivattyúk állapotának értékét egy belső memória változóban tárolja, melyeknek, ha az értéke 1, akkor üzemel a szivattyú, ha értéke 2 akkor pedig hibajelzést generál a program. A 15. ábrán szemléltetem az 1. szivattyúra vonatkozó üzemeltetési és hibajelzési folyamatot, mely a 2. és a 3. szivattyú esetében is ugyan ez. Az összes program hűtésigény jelzését (érlelőkamrák-, klíma-, üzemszerek hűtésigényjelzését) összeszámolja a program, és ez alapján eldönti, hogy hány ammónia szivattyú szükséges a rendszer működéséhez, úgynevezett hűtésigény súlyozást számít. A SCADA-ban beállított értékek alapján fogja eldönteni a program, hogy szükséges-e a két szivattyú üzemeltetése. Egy szivattyú működtetésére mindenképpen szükség lesz, ha van hűtésigény, valamint a SCADA-

ban van lehetőség egy sorrendkapcsoló állítására is, mellyel be lehet állítani, hogy az 1. vagy a 3. szivattyú legyen a prioritizált. A 2. szivattyú automata üzemben csak akkor indul, ha az 1. vagy a 3. szivattyú hibajelzésben van, és szükséges a szivattyú igényszám alapján. A 15. ábrán látható az ammónia szivattyúk hűtésigényének logikai háttére, mely alapján a rendszer kiküldi majd szivattyúknak az üzemeltetés jelet.



15. ábra: Ammónia szivattyúk működésének folyamatábrája, 1 szivattyú által ismertetve

4.2.3. Hibajelek program

Az összes programban kigenerált hibajelzést összegyűjti, és kigenerál 1. szintű hibajeleket és 2. szintű hibajeleket. Ha például a szeletelő üzem egyik ventilátora hibajelzés igaz értékű, akkor 1. szintű hibajelnek számít, ha ugyanezen szeletelő üzem összes ventilátorának hibajelzés értéke igaz, akkor 2. szintű hibajelnek számít. A hibajelek szintje alapján fizikai lámpákat villogtat. Ezen hibajelek a SCADA-ban is mind megjelennek. Bármelyik

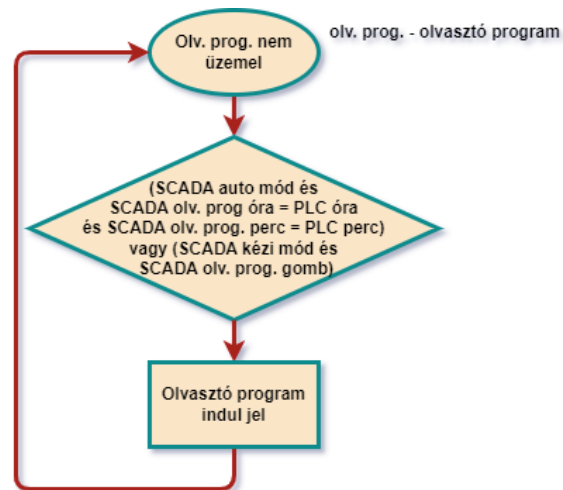
kompresszortól nem érkezik meg az OK jel, vagy az ammóniatartály bármelyik szintjelzése hibát jelez, azt 2. szintű vészjelzésnek generálja.

4.2.4. Kondenzátor programja

A kondenzátor programja a felhasználók elmondása szerint számukra egy olyan része a SCADA-nak, melyet nem szoktak módosítani, az alapbeállítások szerint működtetik. Ebben a részben a kondenzátor ventilátor üzemmódkapcsoló fontos, melyet a SCADA-ban lehet ki, gyors és lassú üzemmódra kapcsolni, és a ventilátorról kapott állapotjelzés, ha nagyobb mint 0, akkor hibajelzést generál a PLC. Ezen kívül a kondenzátor szivattyú üzemmódot lehet a SCADA-ban kézi, automatikus és kikapcsolni, valamint a szivattyú állapotjelzésére is igaz, hogy ha nagyobb mint 0, akkor a PLC hibajelzést generál. A kondenzátor vízfűtés működésére ugyanaz igaz, mint a kondenzátor szivattyúra. A kondenzátor víztálca jelzés és vízhiányra is hasonlóképpen reagál a PLC. A programban ezen kívül egy hosszas rész foglalkozik a kívánt kondenzátornyomás és a mért kondenzátornyomás közötti különbség kezelésével, a gépigényszámításon át a semlegeszóna számításig, melyek számomra nem voltak lényeges részek, a SCADA elkészítése szempontjából. A SCADA-ban állítható értékeket és kapcsolókat elkészítettem és megjelenítettem, de a program belső működésébe nem kellett mélyen beleásnom magam hozzá.

4.2.5. Olvasztóprogram

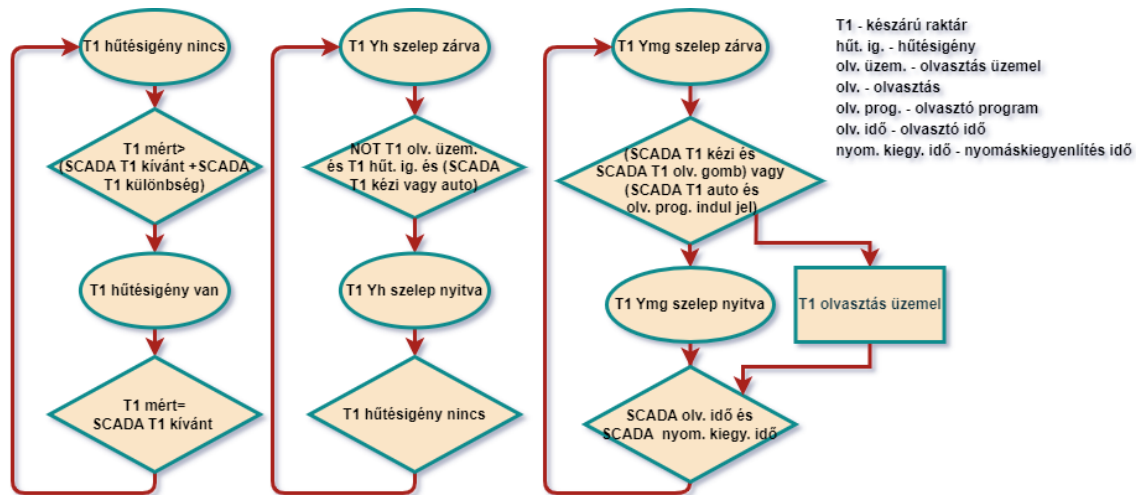
Az olvasztóprogram két üzemmódban üzemel, minden helyiségre külön-külön kézi üzemmódban is elindítható (oly módon, hogy ha kézi üzeme helyezzük az adott helyiséget, akkor megjelenik egy olvasztóprogram indítására szolgáló gomb), valamint előre a SCADA-ban beállított időkből automata módon is tud üzemelni. A PLC-ben eltárolt idő perc és óra értékét összeveti a SCADA-ban beállított olvasztóprogram idő és perc értékét, és ha ezen feltételek igazak, valamint automatikus üzemmódban van, vagy csak szimplán kézi üzemmódban, akkor az olvasztás elindul. Az olvasztás azzal a feltétellel indul meg, hogy az adott rész, szintén automatikus módba kell legyen állítva SCADA-ban. A 16. ábrán látható egy egyszerűsített folyamat ábra. Minden kiadott olvasztó program indul jel az adott helyiségben kiválasztott üzemmódtól függően vezérli a helyiség olvasztóprogramját. Az olvasztó program indul jel a későbbiekben szerepet fog játszani a 4.2.6. fejezetben.



16. ábra: Olvasztóprogram működésének folyamatábrája

4.2.6. Raktár, folyosók és üzemek programja

A T1-től a T6-ig, vagyis a készárú raktár, expedáló üzem, kartonozó üzem, szeletelő üzem, emeleti folyosók és földszinti folyosó esetében ugyanazon program fut, természetesen különböző mért-, kívánt- és különbség értékek alapján, valamint a ventilátorok és szelepek száma változik csak programszinten. A 17. ábrán fogom ismertetni ezen programok folyamatábráját, a T1 készárú raktár példáján keresztül. A mért hőmérsékletértéket elmenti memóriába, és a SCADA-ban beállított kívánt- és különbségérték alapján egy hűtésigény jelet generál. Ha hűtésigény jel van, és nincs olvasztóprogram, valamint automatikus üzemmódkapcsoló aktív a SCADA-ban, nyit a Yh (hűtő folyamathoz tartozó szelep), ha viszont az olvasztóprogram aktív, akkor a Ymg (meleggáz szelep, az olvasztóprogramhoz tartozó szelep) nyit. Az M11 és M12 ventilátorok hűtéskor elindulnak, és egy belső memóriába a PLC elmenti az állapotukat. Ezen belső memória értéke 0, akkor a ventilátor áll, ha 1, akkor üzemel és ha 2, akkor hibajelzést generál. A szelepek állapotát is egy memória változóba menti, 0 akkor szelep zárt állapotban, 1 akkor pedig szelep nyitott állapotban van. A program végén pedig a készárú raktár olvasztóprogramjának további része látható, ahol az automatikus és kézi üzemtől, függően elindul az olvasztás. A 4.2.5.-ös fejezetben tárgyalt Olvasztóprogram által kigenerált T1 olvasztás üzemel jel, ezen programrészben fogja befolyásolni az olvasztóprogram működését a 17. ábrán látható módon.



17. ábra: T1 készárú raktár működésének folyamatábrája

4.3. Tagek kezelése

A programok megértése és a rendszer átlátása után, a SCADA rendszerben lévő változókat rendszereztem, és a fejezetben bemutatott módon úgynevezett tagekként felvettem előre, ezzel is megkönnyebbítve a továbbiakban a munkám.

4.3.1. Tagek felvétele a Gateway-en

A 3.3.2. fejezetben említett wizcon tag listát első lépésként kiválogattam. Azon tagek melyre szükségem volt, egy külön Excel táblázatba gyűjtöttem, oly módon, hogy a megfelelő formátumban írtam őket listaszerűen. A formátum melyhez tartottam magam 18. ábra a) részén látható, az A oszlopban vesszőkkel elválasztva a következő értékeket kellett megadnom: tag neve (Browse Path), címzése (Address), leírás (Description). Az elkészült Excel táblázatot beimportáltam a Gateway-en lévő, a rendszergazdák által előre elkészített, Device Connection megfelelő részére, mely a 18. ábra b) részén látható. Itt lehetőségem lett volna egyesével is megadni az adott értékeket, és felvenni, viszont az Excel táblázat kitöltése, majd importálása gyorsabb megoldásnak tűnt számomra. Omron PLC-nél, nekem az I/O-k megcímezése esetén a CIO előtaggal kellett ellátnom a kívánt tartományt. Belső memóriaértékek esetén pedig a DM előtagra volt a szükség. Holding relay esetén pedig a H betűt kellett írnom. Bit-ek esetén a címzés végére pontot kellett rakni, és utána írni a keresett bit sorszámát (18. ábra a)). Ezen információk a 4.2. fejezetben feltüntetett Omron PLC felhasználói kézikönyvből származnak.

A	
1	T1_SP,DM52,T1 setpoint
2	T1_PV,DM50,T1 process value
3	T1_DV,DM54,T1 deviation value
4	T1_MA,H0.00,T1 automatic mode
5	T1_MM,H0.01,T1 manual mode
6	T1_CD,CIO20.00,T1 cooling demand
7	T1_M11,DM92,M11 engine running

a)

✓ Read completed. [Ignition OPC UA Server]ns=1;s=[AM Gephez 2]CIO285	
Value:	0
Quality:	Good
Timestamp:	2024. 11. 03. 15:58:12 CET
Server:	refresh
Object:	Ignition OPC UA Server
Object:	Devices
Object:	[AM Gephez 2]
Object:	Tags
Tag:	[s][r][w] 1ALARM
Tag:	[s][r][w] 2ALARM
Tag:	[s][r][w] ACKNOW_B
Tag:	[s][r][w] API_A

c)

Device Connections / AM Gephez 2

Load Configuration File			Remove All
Browse Path	Address	Description	
T1_SP	DM52	T1 setpoint	Remove
T1_PV	DM50	T1 process value	Remove

b)

18. ábra: Tagek felvétele a Gateway-ben, a) importálásra került Excel táblázat egy részlete, b) Device Connection menüpont c) felvett tagek ellenőrzése a OPC Quick Client menüpontban

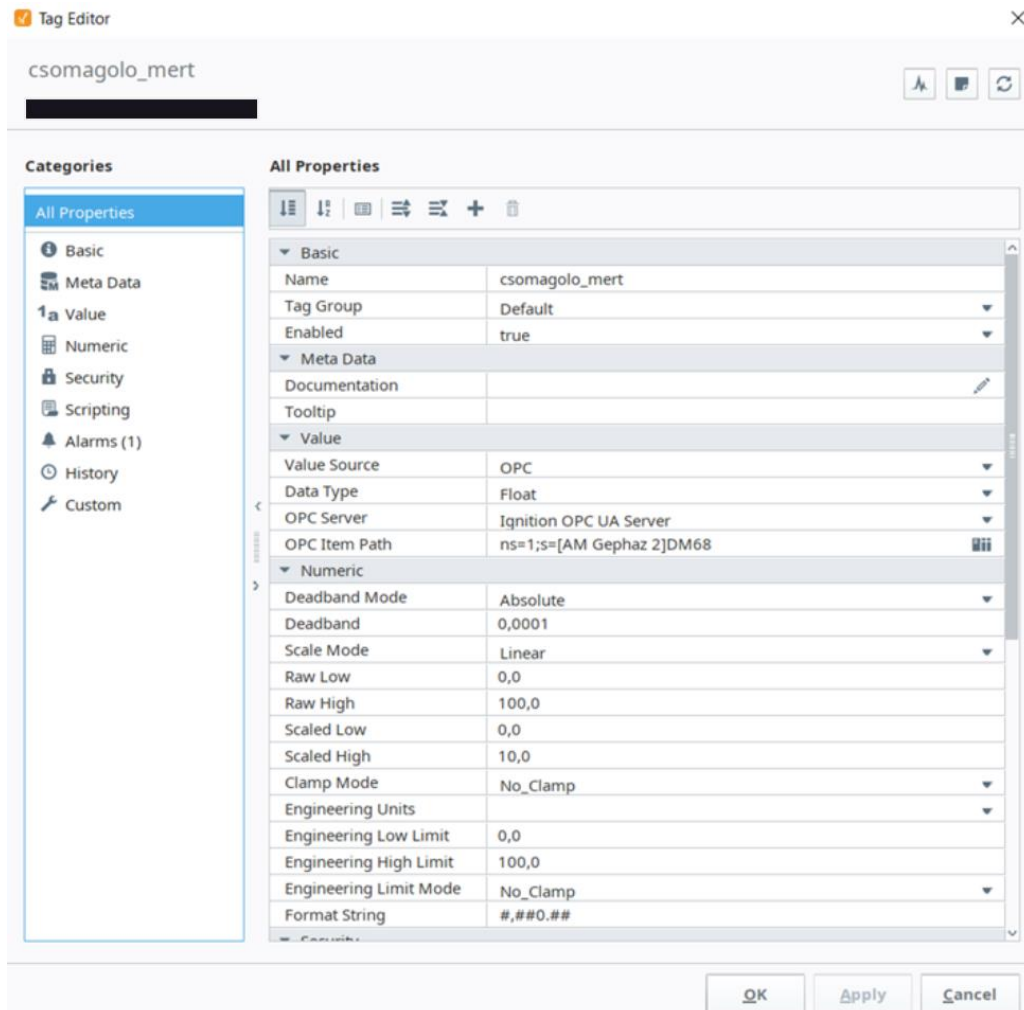
A tagek felvétele után lehetőségem volt az OPC Quick Client menüpont alatt ellenőrizni a tageket, mely a 18. ábra c) részén látható. A felvett tageket itt lehet olvasni (read) és írni (write). Olvasás esetén az értékük, és a hálózati kapcsolat minősége látható, valamint a pontos címezésük, melyet majd a Designer-ben meg kell adni.

4.3.2. Tagek formázása a Designer-ben

A Gateway-en felvett tageket a Designer-ben nekem tetszőleges módon csoportosítottam, és mappákba helyeztem, összesen 152 taget vettem fel a PLC-ből, melyek mind úgynevezett OPC tagek (olyan tagek, melyeknek az értéke a hálózati csatlakozásból érkezik). Ezen kívül készítettem pár Memory (memória) és Expression taget (valamilyen megírt összefüggés, kifejezés alapján alakul ki a tag értéke) is.

A Tag Editorban meg kellett adni a Basic beállításokban a tag nevét. Value beállításokban megadtam a tag típusát, OPC tag esetén a Gateway-ben látott pontos címezésüket, és hogy a tag melyik OPC szerveren fut. Numeric beállítások között az értékek skálázása volt lehetséges, Security beállítások között lehetőség volt a tag írás és olvasás védettségének beállítására. Ezen kívül 3 fontos beállítási rész van még, a Scripting, Alarms, és a History, melyekről a későbbiekben fogok említést tenni. A 19. ábrán látható a csomagoló üzemben mért hőmérséklet

érték Basic, Value és Numeric beállításai, valamint a Tag Editor. Az érték a 3.1.2. fejezetben bemutatott ellenálláshőmérőből származik, és ez alapján skáláztam.



19. ábra: Tag Editor

A History beállításokat engedélyezni kellett azoknál a tageknél, melynek értékeit szerettem volna elmenteni adatbázisba. Storage Provider-nek a rendszergazdák által előre elkészített elérést adtam meg, mely direkt a hűtőgépház SCADA-jának készült adatbázis. Utána beállítottam, hogy milyen rendszerességgel és mikor mentsen értéket az adatbázisba. Lásd: 1. melléklet.

Az Alarms beállítások részt azon tagek esetén kellett beállítani, mely tagek értékéhez valamilyen riasztást szeretnénk állítani. Új riasztás hozzáadására a pluszjel ikonnal volt lehetőség, és utána rengeteg beállítás jelent meg, a riasztás nevétől kezdve, addig, hogy a tag értékének milyen esetében történjen a riasztás. Lásd: 1. melléklet.

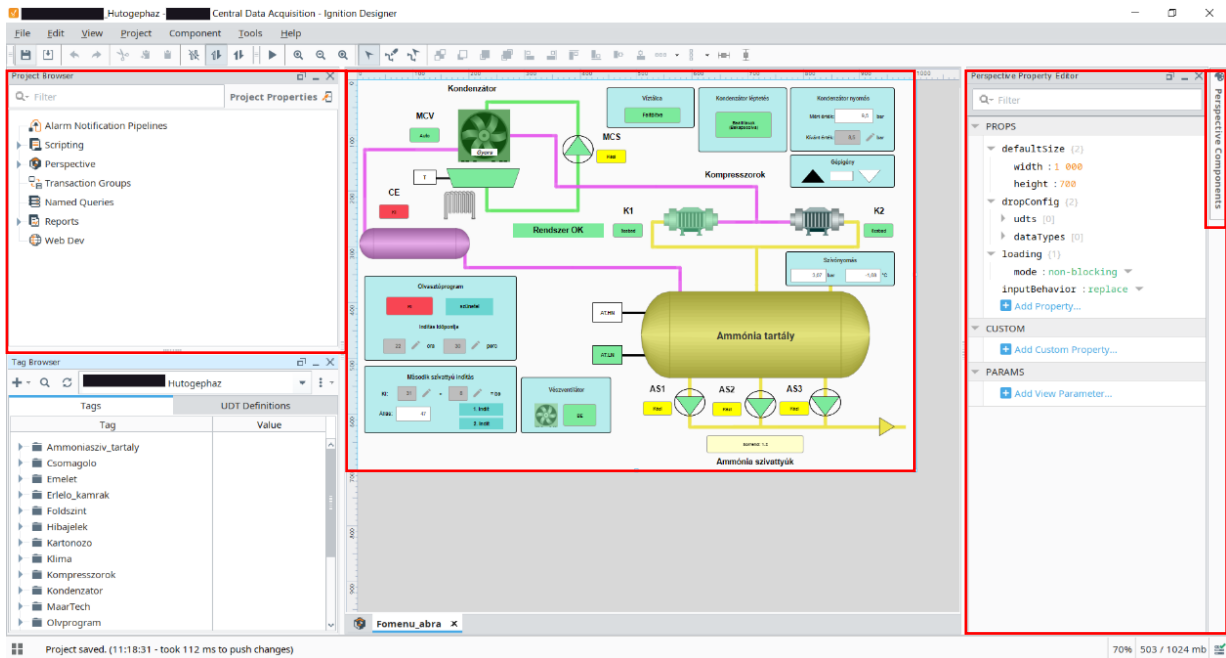
A Scripting beállítások között volt lehetőség scriptek megírására, az 1. mellékleten látható utolsó példán az adott, vagyis scriptelt tag értékének változására, egy másik tag, jelen esetben

a nyugta nevű tag értéke 0-ra vált. Ezen példa azért volt fontos, mert a SCADA-ban megnyomott nyugta gomb egy nyugta nevű változót 1-re írt át, és a PLC-től érkező másik jel állította vissza 0-ra, így megbizonyosodva arról, hogy a nyugtázás megtörtént.

A 2. mellékletben pedig egy másik példát mutatok be a tagek scriptelésére, melyre azért volt szükség, mivel a PLC-be BCD kódolt értéket kellett küldeni és kiolvasni sok tag esetén, viszont a felhasználó az értéket a SCADA rendszerbe decimális értékként adja meg. A szoftver nem biztosított számomra könnyű megoldást, így készítettem egy úgynevezett Memory taget (nyomaskiegyenlitesido_memoria), melynek az értékét a SCADA megjelenítőfelületén tudnak a felhasználók beírni decimális értékben, és ezen felül felvettem az OPC taget (nyomaskiegyenlitesido), ami pedig a PLC-ből érkező BCD érték volt. Mind a két tagre írtam egy scriptet, Value Change (érték változás esetén fut le a script) módon, amiben az oda vissza kódolást Python alapú megoldása látható.

4.4. Megjelenítő felület elkészítése

A megjelenítőfelület elkészítése teljes egészében a Designer-ben történt. A 3.3.1 fejezetben tárgyalt SCADA képernyőfotók és a felhasználók által kért módosítások alapján dolgoztam. A Designer felülete a 20. ábrán látható, melyből kiemelném a Project Browser-t, ahol úgynevezett View-kat hoztam létre, és helyeztem el rajtuk a különböző komponenseket. A komponensek, például szövegmező, képek, gombok behúzása a Perspective Components menüből lehetséges. A komponensek módosítását a Perspective Property Editor menüben lehet elvégezni, ahol a komponensek különböző tulajdonságait lehet állítani. Ezen állítható tulajdonságok közül az egyszerűbbek a szín, érték, méret. Viszont minden View-ra behúzott komponensnek vannak sajátos tulajdonságai is. Például egy grafikon komponens esetén a tengelyek megnevezése, tengelyeken megjelenő értékek hivatkozása, színe is beállítható. A beállítás történhet direkt érték megadásával, vagy úgynevezett Binding-olással, amikor az adott komponens tulajdonságának értékét valamilyen Expression, tag vagy akár adatbázis alapján változtassuk, feltételek és megírt kódok alapján. Ezen kívül az adott komponensekre lehet külön Scripteket írni, vagy úgynevezett Eventeket készíteni.



20. ábra: Designer felülete

A 3. mellékletben látható a Projekt Browser menü, ahol elkészítettem a View-kat. Az ábrák és értékek mappában készítettem el azon kijelezni való felület egységeket, ahol ábrák (szelepek, ventilátorok) szerepelnek (ezen komponensek állapotának értékeit, csak megjeleníteni kellett a PLC-ből kiolvasott értékek alapján), az értékek elnevezésű View-kon láthatóak azon értékek, melyek állíthatóak, például kívánt hőmérséklet, nyomógombok (auto, kézi, ki). Az Exportálás és Grafikon mappában azok a View-k helyezkednek el, melyek a Pdf és Excel exportálás, valamint a grafikonok kirajzolásához voltak szükségesek, erről 4.5.1 és 4.5.2. fejezetekben bővebben beszélek. A Popup mappa tartalmazza a felugró úgynevezett Popup felület egységeket, ezen valamilyen Event, vagyis esemény (gomb megnyomása) hatására felugranak a felületen. És végül látható a lista legalján a 4 fő eleme a SCADA felületének, a Footer (láb), Header (fej), Hibalista és Main (fő). A Footer és a Header, a cégben használt sablonok beimportálásával és a cím átírásával hoztam létre, ezek láthatók fent és lent a kijelzőn. A Hibalistáról a 4.5.4 fejezetben bővebben beszélek. A Main részben pedig az elkészített ábra és értékek mappában lévő elemeket hívtam be egy úgynevezett Tab Container használatával. A Tab Container egy olyan View, melynek a felső részén tetszőleges számú, kattintással kiválasztható részre oszthatjuk a felületet. Az úgynevezett aktív Tab, vagyis a kiválasztott részt fogja megjeleníteni a kijelzőn, így tudunk könnyen váltogatni a felület részegységei között.

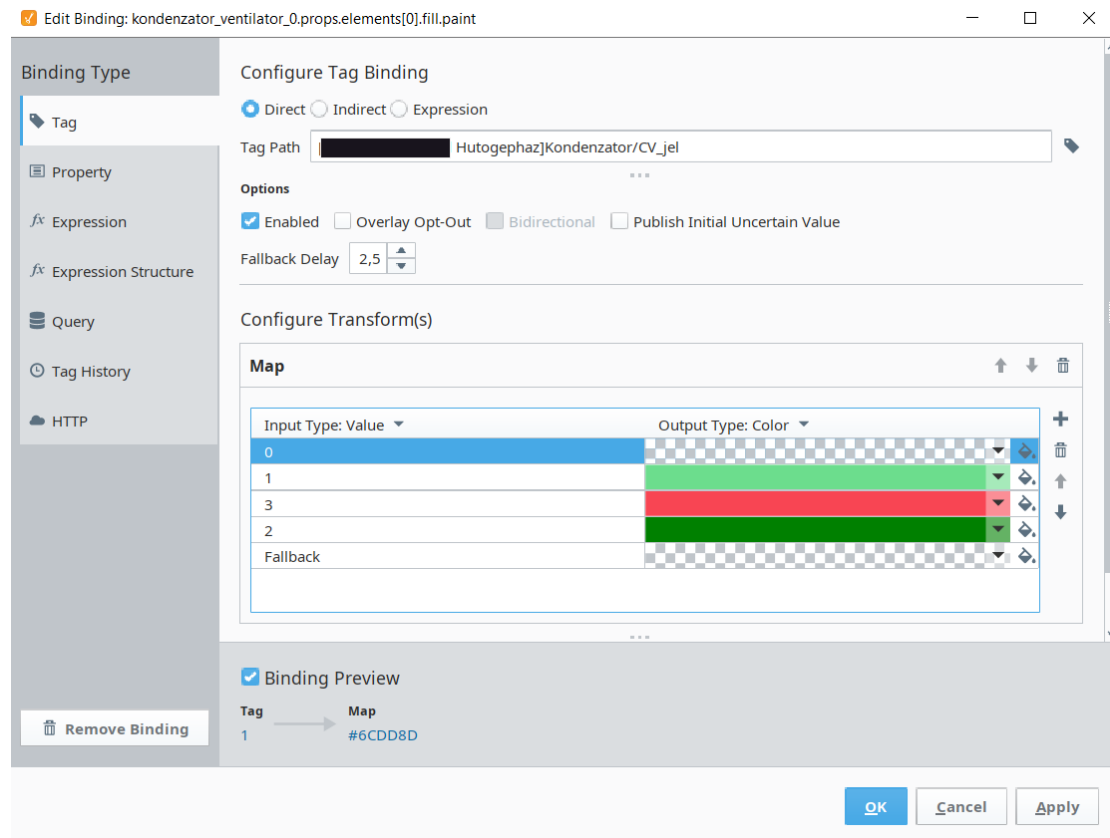
A 3. mellékletben a Projekt Browser menü mellett látható a Perspective Property Editor, mely egy szelep komponens állítható értékeit mutatja be. Az Ignition-nek van egy úgynevezett Symbol Factory tool-ja (eszköztárja), ahol előre elkészített komponensek állnak

rendelkezésünkre, kizárólag ipari eszközökről, például: lámpák, ventilátorok, szelepek, tartályok, motorok. Ebből a Symbol Factory tool-ból származik az adott szelep is, melynek láthatóak az állítható értékei. A 3. mellékletben látható, hogy az elements-en belül a 0. element fill értékének a paint értéke előtt egy kék ikon látható. Ez azt jelenti, hogy az az érték Binding-olva van, tehát valamilyen megadott parancs által változik az értéke (erről a 4.4.1. fejezetben bővebben). Ezen kívül itt állíthatók még a komponens pozíció értékei. A View melyben elhelyeztem ezt a komponenst, percent módban van, vagyis a beírt x, y, magasság és szélesség értékek, nem fix méretűek lesznek, hanem a megjelenítőfelület méretéhez fognak igazodni arányosan. Tehát ha a webes felületet kicsinyítem, nyújtom vagy nagyítom, akkor a komponensek ezzel arányosan x vagy y pozícióban, magasság és szélesség értékben igazodnak a módosításhoz.

4.4.1. *Komponens Binding-olása*

Binding-olás esetén, ahogy az előbb is említett módon adott komponens tulajdonságának értékét valamilyen Expression, tag vagy akár adatbázis alapján változtassuk, feltételek és megírt kódok alapján. A 21. ábrán látható a felugró ablak, amelyben a módosításokat el tudjuk végezni. Ezen az ábrán látható példa esetén a kondenzator_ventilator elnevezésű ventilátor komponens szín értékének Binding-olása látható. A Binding típusa Tag, vagyis egy tag értéke alapján változik, és a Configure Transform(s) (transzformáció beállításai) résznél látható, hogy milyen értékek esetén milyen szín értéke legyen a ventilátornak. Az 21. ábrán látható, hogy a kondenzator_ventilator elnevezésű ventilátor komponens szín értéke változzon, a Kondenzátor mappában lévő CV_jel nevű tag értékére a következő módon:

- hogy ha a tag értéke 0 vagy fallback (nincs jel), akkor a színkód HEX-ben megadva FFFFFFF00
- hogy ha a tag értéke 1, akkor a színkód HEX-ben megadva 6CDD8D
- hogy ha a tag értéke 2, akkor a színkód HEX-ben megadva F84553
- hogy ha a tag értéke 3, akkor a színkód HEX-ben megadva 008000



21. ábra: Edit Binding

A 21. ábrán Edit Binding felugró ablakban látható, hogy nem csak Tag típusú Bindingokat lehet létrehozni. Property típus használatánál, egy másik komponens, egyik kiválasztott értéke alapján lehet módosítani, Expression típus esetén egy megírt programkód, lefutása történik, Query esetén pedig adatbázisból tudunk dolgozni. Ezen kívül a Configure Transform(s) résznél is Map-oláson kívül van még lehetőség más félé transzformációk elkészítésére. Természetesen a feladathoz mérten kell kiválasztani mind a Binding típustát, mind a transzformáció típusát.

A 4. mellékletben látható három másik példa Binding-okra. Az első példában egy Expression típusú Binding látható, ahol az emelet1_41 nevű text komponenst módosítom annak függvényében, hogy az Érlelő kamrák nevű mappában lévő erlelo_42 nevű tag értéke, ha 1, akkor a komponens értéke legyen „42. (1.2) hűtésigény”, viszont, ha nem teljesül a feltétel, a komponens értéke legyen „42. (1.2) szünetel”.

A 4. mellékletben lévő második példán egy Tag típusú Binding látható, egy Format nevű transzformációval. Ebben az esetben a szel_mert nevű text komponens értékét egyenlővé teszem a Szeletelő mappában lévő szeletelo_mert nevű tag értékével, és a transzformációban elvégzett beállításokban a kijelzett értéket tudom formálni. Jelen esetben nekem Numeric, vagyis szám értékre volt szükségem. A Locale-ben beállított auto pedig a szerveren beállított

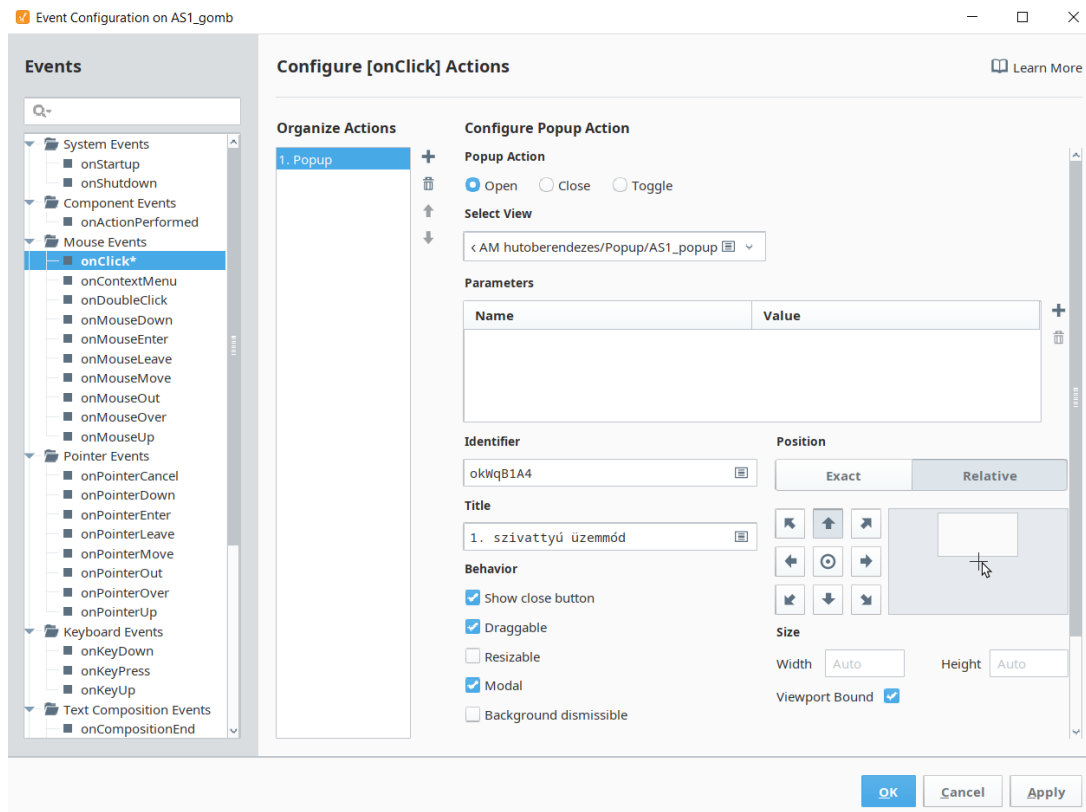
alapértelmezett megjelenítési móddal teszi egyenlővé, vagyis a kijelzett szám érték pont helyett vesszővel legye ellátva, és 1 tizedes pontossággal jelenítsen meg.

A 4. melléklet utolsó példájában a Bidirectional beállítás fontosságát szeretném szemléltetni. Ez a beállítás olyan komponensek esetén lehetséges, ahol módjában áll a felhasználó számára az érték bevitel. A Bidirectional opció kipipálásával nem csak kijelzi az adott komponens a kért tag értékét, hanem felhasználó által beírt értéket egyenlővé teszi a tag értékével, vagyis lehetővé teszi az oda-vissza kapcsolatot a tag és a komponens tulajdonságának értéke között.

4.4.2. *Komponens Skript-elése, Event-ek készítése*

A komponensekre jobb klikkel kattintva előjön egy menü, ahol különböző módosítások elvégzése lehetséges, mint például a törlés, másolás, előre hozás, hátra küldés. Ezeken kívül van lehetőség a komponensre scriptet írni (Configure Scripts), valamint eseményt készíteni (Configure Events). 4.5.2. fejezetben a komponens Scriptelésére lesz példa mutatva, ahol a gomb megnyomására lefut egy Script, mely a kiválasztott adatokat a kiválasztott időintervallum szerint Excel táblázatba rendezi az adatbázisból és letölti a felhasználó által használt eszközre.

A 22. ábrán látható az Event Configuration felugró ablak, melyben lehetőség van eseményeket rendelni az adott komponenshez. Az ábrán látható példában az AS1_gomb nevű nyomógomb típusú komponens megnyomására/kattintására egy felugró ablakként megjelenik a Select View részben beállított AS1_popup View. Az Events listában láthatók az elérhető események, a bemutatott példában a Mouse Event-ek közül onClick esemény van kiválasztva, vagyis kattintásra, vagy érintőképernyős felhasználás során érintésre fog megtörténni az esemény. Organize Actions résznél a pluszjel megnyomásával tudunk eseményeket felvenni, mint például a Popup (felugró ablak), Fullscreen (teljes képernyő mód), Refresh (frissítés), Login (bejelentkezés). A 22. ábrán látható továbbá a Popup Action beállításai, Open esetén a kattintásra a kiválasztott Popup megnyílik, Close esetén pedig bezárja a kiválasztott Popup-ot. Tudunk a Popup-hoz ezen kívül paramétereket rendelni, megnevezni és a megnevezését a felugró ablaknak megjeleníteni, valamint tudjuk pozícionálni, hogy hol és mekkora formában jelenjen meg a kijelzőn. Behavior beállítások között pedig plusz funkciók közül tudunk választani, mint például, hogy a Popup-on jelenítsen-e meg bezárásra alkalmas gombot.



22. ábra: Event Configuration

4.5. Funkciók elkészítése

Miután az összes taget felvettem, elkészítettem a megjelenítőfelület kinézetét (View-kat), és az összes komponenst Binding-oltam, Scriptet írtam rájuk, vagy Event-ként valamilyen eseményt rendeltem hozzájuk, és a tesztelés során felmerülő hibákat kijavítottam, lényegében kimondható volt, hogy a SCADA rendszer önállóan működik, megjelenítésre és a rendszer vezérlésére alkalmas. Viszont az egyéb funkciókat, melyekre szükség volt, még nem látta el. Ebben a fejezetben ezen funkciók elkészítését fogom ismertetni.

4.5.1. Grafikonok kirajzolása

A grafikonok kirajzolásához szükség volt arra, hogy a 4.3.2. fejezetben ismertetett módon a Tag Editorban azok tagek, melyeket grafikonon ábrázolni szeretnénk, History beállításait elvégezzük. Ezen kívül előkészítettem a 4.4.2. fejezetben leírtak szerint azt, hogy adott komponensek kattintása esetén (Configure Event, onClick, Popup, Open beállításokkal), egy Popup ugorjon fel, melyen a grafikon kirajzolódik. Minden Popup Event-hez hozzá rendeltem 3 paramétert (mode, tag_path, tag_path2) és ezekhez egy meghatározott értéket. Azért volt szükség erre, hogy ne keljen minden egyes grafikon külön elkészíteni külön View-n, hanem egy View-n elhelyezett úgynevezett PowerChart komponens paramétereit tudjam változtatni az

Event során változó paraméterek alapján, így mindig az aktuálisan kiválasztott érték jelenjen meg a grafikonon. Az Event során létrehozott paramétereket hozzá adtam a View paramétereikhez, melyen a PowerChart került elhelyezésre.

Ezután a PowerChart tulajdonságainak beállításai következtek, melyet a 5. és a 6. mellékletben látható a PoweChart-hoz Perspective Property Editor és a Bindingok alapján ismertetek. Az 5. mellékletben látható a PowerChart komponenshez tartozó tulajdonságok. Ezen tulajdonságok nagyon széles körűek, betudjuk állítani, hogy milyen időközönként vegyen fel a grafikonon új pontot, melyik adatbázisból dolgozzon, milyen funkciók legyenek elérhetők a felhasználó számára. Vizuálisan szintén minden állítható, tengelyek neve és címe, skálázás léptéke, használt karakterek mérete és még sok más.

Az axes, vagyis tengelyeknél létrehoztam két elemet, 1. és 0. elem. A 0. elem beállításai az 5. mellékletben láthatóak. Ez a 0. elem a nyomás értékek tengelye, az 1. elem pedig a hőmérséklet értékek tengelye. A felvett paraméterek közül a mode nevű paraméter fogja meghatározni, hogy melyik tengely jelenjen meg. Ha a mode értéke 1 akkor a nyomás, ha pedig 2 akkor a hőmérséklet.

Az 5. mellékleten látható ezen kívül a pen-s, vagyis ceruza tulajdonsága a komponensnek, mindegyik ceruza kirajzol egy adott függvényt a PowerChart-on. 3 darab pen-t készítettem, a 0. pen a nyomásértékek esetén fog megjelenni, 1. pen a hőmérsékletértékek esetén és a 2. pen pedig kirajzolja a kívánt nyomás/hőmérséklet értéket. A 6. mellékletben látható az, hogy milyen Bindingokkal oldottam meg ezt a feladatot. A 0. és az 1. pen nevéhez hozzá rendeltem a melléklet első példájában látható módon Property Binding-al a View paramétereik közül a tag_path értéket, a 2. pen esetén pedig ugyanilyen módon a tag_path2 értéket. Ezek az értékek alapvetően a már említett módon komponensre kattintva Event-ként íródnak felül. Például a Csomagoló mért hőmérsékletértékre kattintva, a mode értéke 2, a tag_path értéke Csomagolo/csomagolo_mert és a tag_path2 értéke Csomagolo/csomagolo_kivant lesz. A 6. mellékleten látható utolsó példán egy Expression Binding van, ahol a pen data.source (adat elérés) tulajdonságát változtatom a következő módon. A 0. pen mint említettem a nyomásértékek kirajzolásáért felelős, tehát ha a mode értéke 1, akkor a példában látható adat elérést valósítsa meg (tag_path alapján), ha pedig nem teljesül, vagyis a mode értéke nem 1, akkor 0 kerül beírásra, vagyis ne jelenítsen meg semmit a pen. A 1. pen esetében, ami a hőmérsékletértékek megjelenítéséért felelős ugyanígy jártam el, csak a mode értéke 2-vel kellett, hogy egyenlő legyen. Kívánt hőmérséklet vagy nyomás értékek, vagyis a 2. pen esetében, mode értékétől függetlenül a tag_path2-höz tartozó adat elérést kellett kigenerálnia az elkészített Expression Binding által.

4.5.2. Exportálások megalkotása

Az exportálások elvégzéséhez külön létre hoztam egy View-t, melyen 2 gomb volt (ezen gombok megnyomására történt meg az Excel tábla letöltése, vagy a Pdf-előnézetének megjelenítése), valamint két DateTimePicker komponens, melyekkel kiválasztható az időintervallum. Ezeken kívül volt egy Dropdown komponens, ami egy lenyíló lista formájában kiválaszthatóvá tette a kívánt értéket a Pdf exportálásához. Az Excel táblázat értékeinek kiválasztásához CheckBox komponenseket helyeztem el, hogy egyszerre akár egy Excelben több érték is kimenthető legyen.

Excel táblázat exportálása

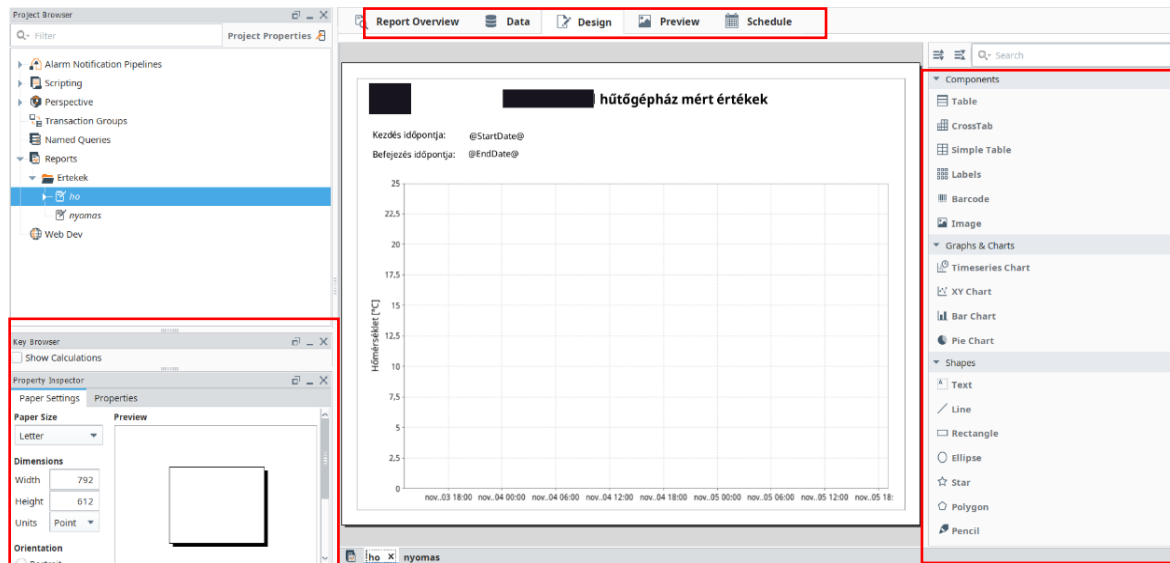
A Excel táblázat exportálásához egy gomb komponensre írtam egy Script-et, ami akkor fut le, ha a beállított Event (onClick) teljesül, vagyis a felhasználó rákattint. A gomb megnyomására lefut egy Script (Lásd: 7. melléklet), mely a kiválasztott adatokat a kiválasztott időintervallum szerint Excel táblázatba rendezi az adatbázisból és letölti a felhasználó által használt eszközre.

A 7. mellékletben látható Scriptben startDate és endDate változókat egyenlővé tettem a DateTimePicker komponensek értékével, majd létre hoztam 3 tömböt, data, tulajdonságok és columnNames névvel. Attól függően, hogy melyik CheckBox aktív, vagyis értéke 1, feltölti a program a 2 tömböt a kért értékekkel, ezután egy for ciklus lefutása következik, ahol a data tömb elemeit tölti fel a már feltöltött tulajdonságok tömb alapján. Az excel1 változót egyenlővé teszi a system.tag.queryTagHistory() függvény alapján, majd ezt a változót a system.dataset.toExcel() függvénnyel Excel táblázatba rendezi, melyet egy excel változóként definiálunk. Egy fajlnev változóban pedig a fájl nevét alakítjuk ki. Végül pedig a system.perspective.download() függvénynek megadjuk a két szükséges változót, a fájl nevét és az excel táblázatot, és ezen függvény lefutásával megvalósul a fájl letöltése a felhasználó által használt eszközre. (A szövegben említett függvények, és a függvény definiálásához szükséges kötelező és opcionális változók a 3.2. fejezetben ismertetett felhasználói kézikönyvből megtekinthetők)

Pdf exportálása

A Pdf exportálásra az Ignition egy külön lehetőséget kínál a felhasználók számára, melyet Report-nak hív. A Projekt Browser-ben a Reports-ok közé létrehoztam egy Ertekek mappába 2 darab Reports-t, egyik neve ho, mely a hőmérséklet értékek kimentéséért lesz felelős, a másik

neve nyomás, mely a nyomás értékek kimentéséért lesz felelős. A 23. ábrán látható a Reports-ok kezelésének felülete.



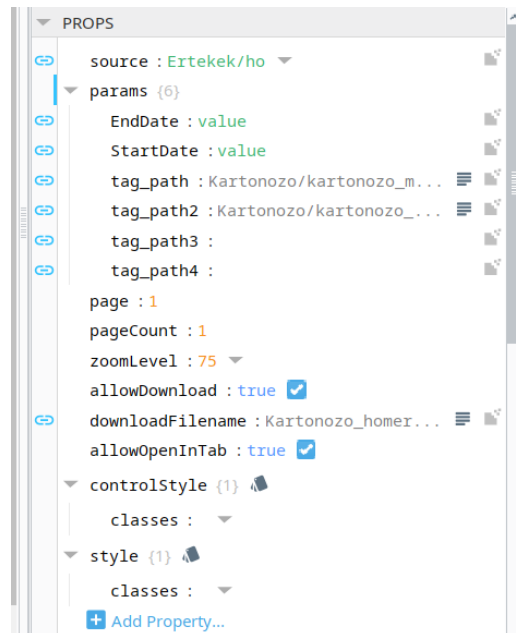
23. ábra: Reports-ok készítésének felülete

Data fül alatt a Report-on megjeleníteni kívánt értékeket tudjuk állítani, Preview fül alatt egy előnézeti képet tudunk látni róla, és a 23. ábrán is látható Design fül alatt pedig el tudjuk helyezni a Report-on a kívánt komponenseket. Az ábrán látható példán 5 darab Text (szöveg), 1 darab Images (kép) és 1 darab Timeseries Chart (grafikon) komponenst helyeztem el a Report-on. Ezen példán a hőmérséklet értékekért felelős ho Report látható, de a nyomás elnevezésű is ugyanezen módon készült el. A Property Inspector menüben tudjuk állítani a kiválasztott komponens tulajdonságait, a 23. ábrán látható maga a Report beállításai, mint például a méret, vagy hogy álló vagy fekvő elrendezésű legyen.

A 8. mellékletben látható a hőmérséklet értéket kijelző Timeseries Chart Property Inspector beállításai. A pirossal kirajzolt kívánt hőmérséklet érték elérési kulcsaként key2, míg a zölddel kirajzolt mért hőmérséklet érték elérési kulcsának key-t adtam meg. A 8. mellékletben látható ezen key és key2 értékeinek származási helye. A Report Data beállítási között látható, hogy 4 paramétert (StartDate, EndDate, tag_path, tag_path2) áthozok View-ról, hogy a Report értékeiként is definiálva legyenek (ezekről későbbiekben szó fog esni bővebben). A Data Sources részben Query-tag history-ként felvettem a key és key2 kettőt, melyek fogják a grafikon számára szolgáltatni a kirajzolni kívánt értékeket a StartDate és az EndDate, tag_path és a tag_path2 alapján (A működési elve hasonló a 4.5.1. fejezetben használt megoldás során, csak itt az értékek átadása másképpen működik). Ezen kívül a 8. mellékletben látható, hogy Data Sources közé egy Script formájában változtatom a Title vagyis a Report cím értékét. A

Scriptben egy tag értékét olvasom ki, melynek neve Pdf, és ez a tag értéke alapján mondom meg, hogy mi legyen a címe a Report-nak.

A megjelenítő felületen található exportálás Pdf-be gomb megnyomása esetén Event-ként beállítottam, hogy felugorjon a Report Graph nevű View, valamint a gomb megnyomásakor az Event paraméterei közé elmentettem az EndDate és StartDate változóba a DateTimePicker komponensek időértékeit. A megjelenítőfelületen található Dropdown komponens értékét, ami egy lenyíló lista formájában kiválaszthatóvá teszi a kívánt értéket, amit a grafikonra szeretnénk megjeleníteni, elmentem Binding-al a Pdf nevű tag értékébe. A Report Graph View-n egy ReportViewer komponens helyeztem el, mely kifejezetten a Report-ok elkészítéséhez és megjelenítéséhez használt komponens. A komponens tulajdonságairól egy részlet a 24. ábrán látható.



24. ábra: Részlet a ReportViewer komponens tulajdonságairól

A 24. ábrán látható source tulajdonságnak meg kell adni, hogy melyik Report-ot jelenítse meg. A példán látható Ertekek/ho, az Ertekek mappában lévő ho nevű Report, melyet egy Expression Binding-al oldottam meg. Ha a Pdf nevű tag értéke 0 vagy egy, akkor Ertekek/nyomas, minden más esetben a source tulajdonság értéke legyen Ertekek/ho. A params tulajdonságoknál felsorolt listában láthatóan azok, melyeket a Report beállításában a Data paramétereiként megadtam (8. mellékelt). EndDate és a StartDate a már említett DateTimePicker komponensek értékeiből származik. A tag_path és tag_path2 beállításához Tag Binding-ot használtam Map transzformációval. A Pdf tag értéke alapján írja felül a változók értékeit. 0 és 1 érték esetén nem ír be semmit a változók értékeibe, hiszen a 0 és 1 Pdf tag érték esetén nyomásértékekről beszélünk. Tag_path3 és Tag_path4 a nyomás nevű Report-

hoz tartoznak, ha a Pdf tag értéke 0 vagy 1, akkor fogják beírni a változók értékét, minden más esetben üresen hagyják, mint a 24. ábrán látható esetben is történik. Az ábrán látható utolsó tulajdonsága a komponensnek, melyet Binding-oltam a downloadFilename, mely megadja letöltés esetén a Pdf nevét, erre egy Expression Binding-ot írtam, hogy kívánt fájl nevet kigenerálja nekem.

4.5.3. Felhasználói szintek kezelése

A felhasználói szintek kezelésének megoldása kulcsfontosságú volt, mielőtt a projekt felkerült az úgynevezett navigációs felületre, ahol mindenki, aki a cégcsoport hálózatára csatlakoztatott eszközzel a link birtokába van, megtudja tekinteni az elkészült SCADA rendszereket, hiszen a SCADA értékeinek átírásával az egyik telephely hűtőgépházának vezérlése lehetséges. A cél az volt, hogy mindenki számára megtekinthető legyen a felület teljes egészében, viszont a következő funkciók legyenek felhasználói szinthez kötve: bármilyen érték átírása, szivattyúk, kompresszorok és más hűtéstechikai elemek ki/ bekapcsolása, olvasztóprogramok elindítása, hibalista törlése és nyugtázása.

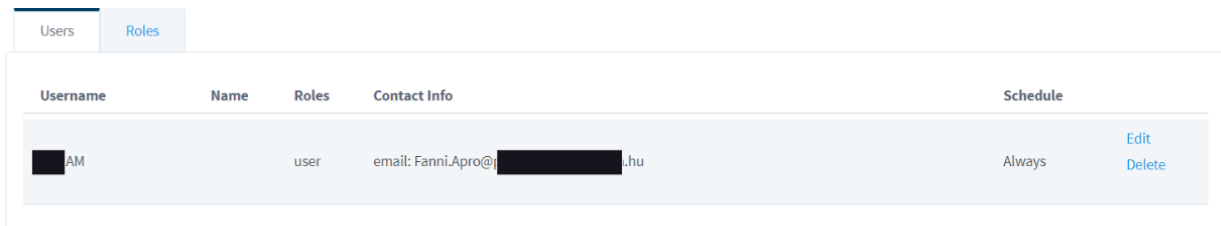
A feladat megoldásához az Ignition egy igen jó megoldással rendelkezik, Button, vagyis gomb komponenst helyeztem el (ki- és bejelentkezésre egyaránt) a Header részére a megjelenítőfelületnek, ahol tudtam Login (bejelentkezés) és Logout (kijelentkezés) Event-et hozzá adni, mely onClick eseményként, vagyis kattintásra következett be. A megfelelő felhasználónév és jelszó páros beírása esetén a bejelentkezés sikeres, és a Header-ben megjelent a felhasználó neve, és a hozzá rendelt felhasználói szint.

Ez említett funkciók letiltásához pedig arra volt szükség, hogy minden komponens enabled (elérhető) tulajdonságára egy Expression Binding-ot készítssek (25. ábra). Ezzel a komponens enabled tulajdonsága false, vagyis hamis értéket vesz fel, kivéve akkor, mikor a felhasználói szint értéke user, abban az esetben true, vagyis igaz értékű lesz. Ezzel például egy nem megfelelő jogosultsággal rendelkező felhasználó számára egy NumericEntryField (számérték bevitelére alkalmas mező) komponens beírt értéke látható, viszont a módosítás nem lehetséges.



25. ábra: Binding a felhasználói szintek kezeléséhez

A Designer-ben elvégzett módosításokon kívül a Gateway-en szükséges volt beállítani a megfelelő felhasználókat, a hozzájuk tartozó jelszavakat, és a Projekthez párosítani ezeket. A Gateway-en létre kellett hozni egy úgynevezett Identity Providers-t (személyazonosság szolgáltató), melyet a Designer-ben a projekt beállításaiiban a projekthez rendeltem. Ezen felül a Gateway-en Users, Roles (felhasználók, szerepek) menüpontban felvettem a hűtőgépház felhasználóinak listáját. Itt a felhasználó név, jelszó, Role-on kívül e-mail címet rendeltem a felhasználóknak. A 26. ábrán látható azon felhasználó, melyet létrehoztam magamnak tesztelés céljából.



Username	Name	Roles	Contact Info	Schedule	
AM	user	user	email: Fanni.Apro@...hu	Always	Edit Delete

26. ábra: Gateway-en történő Users, Roles beállítások részlet

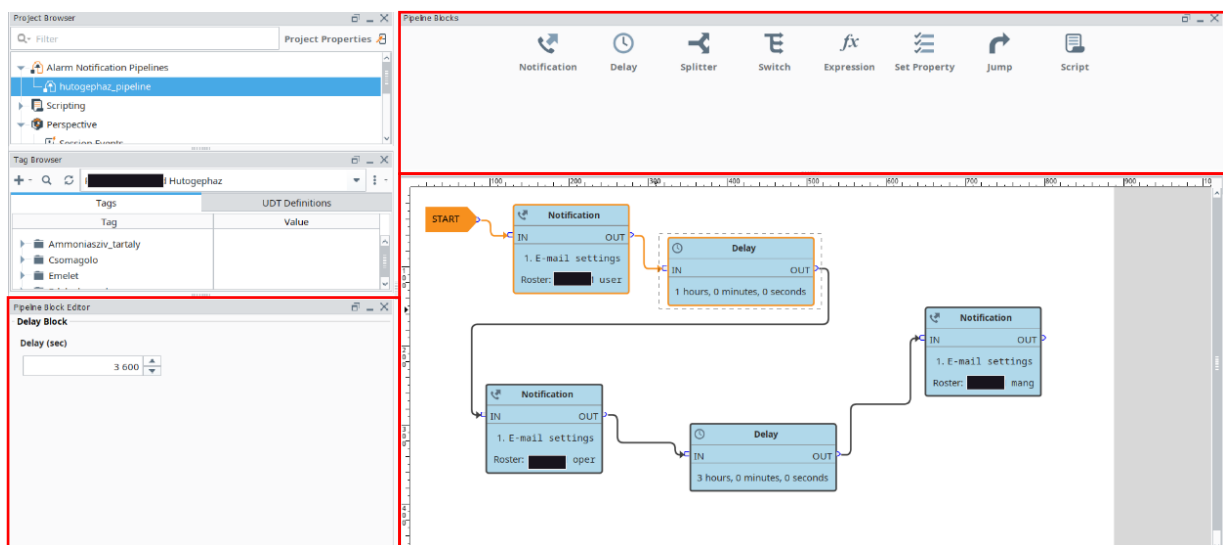
4.5.4. Alarm rendszer kialakítása

Az alarm rendszer kialakításához szükség volt arra, hogy a 4.3.2. fejezetben ismertetett módon a Tag Editorban azon tagek, melyekre riasztást szeretnénk, az Alarms beállításait elvégezzük.

Ezen kívül a Hibalista View-n elhelyeztem a nyugta gomb mellett, egy AlarmStatusTable komponenst, mely a megfelelő beállítások mellett, a Projekten Alarms beállításokkal rendelkező tageket kilistázza. A komponens tulajdonságai között rengeteg módosítási lehetőség állt rendelkezésemre. A különböző típusú alarmok például: diagnosztikai, alacsony, közepes, magad, kritikus szintű hibák, valamint az aktív, már nem aktív, nyugtázott és még nem nyugtázott hibák megjelenítése közötti különbségeket tenni beállításokban vizuálisan. Hibák törlésének lehetősége mellet, a tábla frissítésének idejét, és ezen felül egyéb funkciók engedélyezése volt lehetséges. A táblában az én beállításaim között szerepelt még, hogy az adott hibáról a következő adatokat mutassa meg: riasztás kezdetének időpontja, hiba forrása (melyik tag került hibastátuszba), Állapot (aktív, nyugtázott stb.), Hiba pontos megnevezése (ez a tag Alarms beállításaiiban volt megadható), és az úgynevezett Clear Time, vagyis, hogy a hiba azon időpillanat, amikor a hiba már nem aktív státuszba került át.

Mindegyik fő View-n elhelyeztem egy Hibalista gombot, mely kattintással átnavigál a Hibalista View-ra, és a gombra készítettem egy Expression Binding-ot, hogy bármilyen hiba esetén feltűnő vörös színnel villogjon a gomb.

Az utolsó lépés az Alarm rendszer kialakításában az e-mail küldés hiba esetén volt. A Projekt Browser részen Alarm Notification Pipelines nevű egységbe készítettem egy hutogephaz_pipeline-t, mely az e-mail értesítés rendszerért volt felelős. Az Alarm Notification Pipelines készítésének felülete a 27. ábrán látható. A Pipeline Blocks részen választottam ki a különféle egységeket, melyekből felépíthető volt egy riasztási folyamat. Minden egység beállítása külön a Pipeline Block Editor-ban volt lehetőség. Itt állítottam be a riasztást e-mailra, írtam meg script formájában a szöveg tartalmát, állítottam be, hogy milyen e-mail címről érkezzen a riasztás, valamint, a Gateway-en beállított 4.5.3. fejezetben tárgyalt Users, Roles beállítások alapján azt, hogy melyik felhasználó csoporthoz tartozzon a riasztás, és azon belül melyik felhasználót vagy felhasználókat érintse.



27. ábra: Alarm Notification Pipelines beállítások felülete

5. EREDMÉNYEK

A munkám során elkészült SCADA rendszer eredményeként első sorban a 9-17. mellékletben található képernyőfotókat szeretném kiemelni. Mivel egy működő rendszer vezérlésére és felügyeletére szolgáló megjelenítőfelületet készítettem, így az eredményekről a képernyőfotók megtekintésével lehet a legtöbbet megtudni.

A 9. mellékleten látható a SCADA bejelentkezés előtt és után, ahol a felhasználói szintek kezelésére hívnám fel a figyelmet. Bejelentkezés előtt minden érték látható, szürke mezőben azok, melyeknek módosítása nem lehetséges, ám bejelentkezés után már igen. 10-12. mellékletekben láthatók a főképernyőn kívül megtekinthető oldalak, és egy példa, hogy a grafikon kirajzolására a hőmérsékletértékre kattintva. 13 és 14. mellékletekben a Popup-okra, vagyis felugró ablakokra láthatóak példák, és egy nyomásérték grafikon kirajzolására. 15. mellékletben látható a felület, ahol az exportálásokat végezhetjük el beállított időintervallum és nyomás/hőmérséklet értékek szerint. Valamint a Pdf előnézete látható, ahonnan letölthető a felhasználó által használt eszközre. 16. mellékletben a kimentett Excel táblázat egy részlete és a Pdf exportált formája tekinthető meg. A 17. mellékletben pedig a hibalista gombra kattintva megjelenő Hibalista és a nyugta gomb látható, ezen kívül riasztás esetén az e-mail, melyet a megfelelő felhasználók kapnak szükség szerint.

A projekt eredményének anyagi vonzatát is megemlíteném, melyet külsős cég megbízásával közel 3 millió forint értékében valósították volna meg, viszont így „házon belüli” megoldásként 0 forint költséggel oldottam meg.

A projekt jelenleg lezártnak tekinthető, viszont a telephelyen jelenleg egy üzemrész újra termelésbe hozása esetén, a SCADA könnyen bővíthető vele. Ezen kívül a jövőbeli tervek között szerepel a telephelyen lévő összes műszaki adat megjelenítése, például mérőórák, érzékelők, berendezések. Az elkészült SCADA rendszer projektjében már pár adatkapcsolat és a tőlük származó értékek skálázva már, fel is kerültek a Gateway és a Designer-be, így kimondható, hogy a közeljövőben az elkészült SCADA bővítése a cél, hogy a telephely összes műszaki adatkapcsolatát felügyelje.

Összefoglalás ként a 2. táblázatban látható, hogy a régi SCADA rendszerhez képest az elkészült új SCADA rendszer milyen funkciókkal bővítette a felhasználók lehetőségeit.

2. táblázat: Régi és új SCADA rendszer összehasonlítása

Funkció megnevezése	Régi SCADA	Új SCADA
Időt állóság:	Csak Windows XP operációs rendszerű számítógépre telepíthető	Webserver (internet elérhetőség szükséges)
Elérhetőség:	Gépházban	Bárhonnan a cégcsoport hálózaton
Felhasználói szintek kezelése:	Nem	Igen
Adatok tárolása:	Rövid távú trendek (grafikonok) készítéséhez	3 évig tárolja az adatokat
Grafikonok:	Igen, de nem állítható az időintervallum	Igen, de állítható az időintervallum
Adatok exportálása:	Nem	Excel és Pdf formátumba
Riasztások:	Csak a gépházban tekinthetők meg	E-mail értesítés küldése felhasználóknak, hibalista megtekinthető bárhonnan a cégcsoport hálózaton

6. ÖSSZEFOGLALÁS

A dolgozat témája hűtőgépház PLC és SCADA rendszerének korszerűsítése, melyből adódik, hogy hűtéstechikai alapismeretek megértése elengedhetetlen volt a feladat elkészítéséhez, hogy a rendszert mögött lévő technológiát átlássam. Ezen kívül az ipari automatizálás egyes részeinek elsajátítására is szükségem volt. A telephely hűtésének megvalósítása PLC vezérelt rendszerként működött, így szükséges volt a vezérléstechnika és PLC rendszerek ismerete. A cél viszont egy SCADA rendszer kialakítása volt, így az automatizálási piramis lépcsőfokainak átlátása, és a szintek közötti kommunikáció megalkotásának módjával is megismerkedtem, valamint a SCADA készítés fő lépéseivel.

A rendszer megismerése, meglévő PLC program és egyéb információk feldolgozása után, a legtöbb időt a SCADA tervező szoftver használatának megtanulása vette igénybe. Az Ignition egy nagyon széleskörben alkalmazható szoftver, mely sok funkció megalkotására kínál lehetőséget. A felhasználói kézikönyve, mint a szoftvernek, mint a PLC-nek a munkám során végig rengeteg segítséget nyújtott. A szoftver Python programozási nyelven alapul, így a Python programozás alapismereteit is volt szerencsém elsajátítani.

Mivel a SCADA egy működő és meglévő rendszert vezérel, egy teszt szerveren próbáltam ki az elkészült funkciókat, tesztelés szempontjából, mielőtt a főszerveren is megalkottam volna ezeket. Az éles tesztelések során, pedig megtanultam, hogy milyen lépésekre kell figyelni, és hogy nagy segítséget tud nyújtani, ha a teszt során a rendszert nap mint nap felügyelő és működtető kolléga mellettem van, aki probléma esetén tud segíteni. Megtanultam, azt is, hogy a fő teszt után érdemes néha-néha vissza tekinteni a projektekre és a felhasználókkal konzultálni, hogy észleltek-e problémát vagy hiányosságot működés közben, melyet utólag tudok korrigálni.

A munkám célja egy olyan SCADA rendszer létrehozása volt, mely a külső és belső szabványoknak megfelel. Valamint azon kollégáknak, akik nap mint nap felhasználói a rendszernek, megfeleljen. Végeredményben a céloom elérése sikeresnek tekinthető, minden tervezett és elvárt funkciót sikerült elkészíteni, és a felhasználók elégedettek a megvalósult rendszerrel.

IRODALOMJEGYZÉK

- Ajtonyi, I. (2007). *PLC és SCADA-HMI rendszerek I.* Miskolc: AUT-INFO Kft.
- Ajtonyi, I. (2008). *PLC és SCADA-HMI rendszerek II. & Ipari kommunikációs rendszerek II.* Miskolc: AUT-INFO Kft.
- Basant, T. & Narendra, K. (2020). PLC and SCADA based Industrial Automated System. *IEEE International Conference for Innovation in Technology.* Nov 6-8, 2020
- Beke, J. (1994). *Hőtechnika a mezőgazdasági és az élelmiszeripari gépészetben.* Budapest: Agroinform Kiadó és Nyomda Kft.
- Boyer, S. A. (2009). *SCADA: Supervisory control and data acquisition.* Amerikai Egyesült Államok: International Society of Automation
- Csikós, S. (2014). PLC programozás. Szegedi Tudományegyetem, Mérnöki Kar. 2024. 11. 01.
- Fekete, I. & Menyhárt J. (1965). *A légtechnika elméleti alapjai.* Budapest: Műszaki Könyvkiadó.
- Gänger, Gy. (1964). *Hűtőipari technológia I.* Budapest: Műszaki Könyvkiadó.
- Gänger, Gy. (1966). *Hűtőipari technológia II.* Budapest: Műszaki Könyvkiadó.
- Hugh, J. (2007). *Automating manufacturing Systems with PLCs.* Letöltés dátuma: 2024. 11. 01.
- Husi, G. (2016). *Ipar 4.0.* Debreceni Egyetem Műszaki Kar. Letöltés dátuma: 2024. 10. 30.
- Jakab, Z. (2006). *Kompresszoros hűtés I.-II.* Budapest: Hűtő és Klimatechnikai Vállalkozások
- Kiss, É. & Tiner, T. (2021). Az ipari forradalmak és az infokommunikációs fejlődés földrajzi összefüggései a nemzetközi szakirodalom tükrében. *Földrajzi Közlemények.* 2021. 145. 2. pp. 89–105
- Láng, L. & Jakab, Z. (1984). *Hűtéstechika.* Budapest: Műszaki Könyvkiadó.
- Rácz, L. (2011a). *Hűtési rendszerek, hűtőberendezések.* Szeged: Belvárosi nyomda.
- Rácz, L. (2011b). *Légtechnika, klimatechnika.* Szeged: Belvárosi nyomda.
- Thornhill, C. J. & Lowrey, B. A. (1993) PLC control of a cold store and ripening plant - a case history. *The Institution of Electrical Engineers.*
- Tuhsnajt, M. V. & Emblik, ED. (1956). *Élelmiszerek hűtése és fagyasztása.* Budapest: Műszaki Könyvkiadó.
- Vasáros, Z. (2020). *A termodinamika alapjai.* (L. Juhász, T. Maiyaleh, J. Vadász & Z. Vasáros, szerk.) Budapest: Nemzeti Klímavédelmi Hatóság. Letöltés dátuma: 2024. 10. 13.